MASTER EN SCIENCES INFORMATIQUES

Méthode de définition des paramètres fixant les tampons dans un réseau de télé-processing

Poncelet, Jean-Marc

Award date:
1979

Awarding institution:
Universite de Namur

INSTITUT D'INFORMATIQUE
Facultés Notre-Dame de la Paix
NAMUR

Méthode de définition des paramètres fixant les tampons dans un réseau de télé-processing

PONCELET Jean-Marc

Directeur de mémoire: M.J.RAMAEKERS
Mémoire présenté pour l'obtention du grade de Licencié et Maître en Informatique

Nous tenons à remercier vivement IBM pour la possibilité qu'il nous a offerte d'effectuer un stage de cinq mois au sein de son entreprise ainsi que pour l'accueil et le comportement de ses membres.

Plus particulièrement, nous tenons à remercier M.MEURET qui a fait les démarches nécessaires pour permettre ce stage, M. BRASSEUR qui a opéré nombre de démarches afin de rendre l'élaboration de ce mémoire possible et MM. LEBOUTTE et NACHTERGAEL pour leur aide précieuse lors des problèmes techniques.

Nous sommes également reconnaissant à MM.BERTHILIER et MOURMAUX qui nous ont parmis de disposer des données nécessaires.

A L'Institut d'Informatique, nous remercions M. RAMAEKERS, promoteur,et M. PAULUS qui nous ont suivi par leurs conseils tout au long de l'évolution de ce travail.

TABLE DES MATIERES

oooooooo

I N T R O D U C T I O N

==========

Ces dernières années, le nombre et la complexité des fonctions assumées par le système se sont accrus jusqu'à donner le dernier-né des systèmes d'exploitation: MVS (multiple virtual storage).

De même, l'utilisation, à distance, des facilités de l'ordinateur et de l'envoi de données, ont pris un essor particulier.

Dans l'ensemble des tâches du système, il en existe une plus particulièrement dont la fonction est d'assurer le contrôle et la transmission des données à distance : VTAM.

La transmission de ces données nécessite la présence de zones de transit qui seront donc destinées à prendre en charge, temporairement, ces données jusqu'à ce qu'elles soient transmises, soit vers les terminaux, soit vers les processus du processeur central.

Plus particulièrement, les paramètres qui définiront ces tampons ont une importance primordiale dans le bon fonctionnement du réseau.

En effet, une mauvaise définition aura des conséquences graves sur le comportement de VTAM.

Or, les méthodes proposées par IBM, pour la définition de ces paramètres, présentent quelques lacunes.

C'est pourquoi nous avons tenté de donner un solution pratique aux problèmes posés par la définition de ces

paramètres.

Notre objectif sera de déterminer la taille mémoire minimum nécessaire pour éviter le ralentissement.

A cette fin, nous construirons un simulateur d'allocateur des tampons qui dégagera les informations nécessaires.

Ce simulateur fonctionnera à partir de données fournies par VTAM à un outil de mesure.

Ces données regrouperont des renseignements sur tout ce qui utilisera les tampons de VTAM.

Elles permettront de simuler l'allocation aisément.

La libération nécessitera le calcul d'un paramètre propre à chaque longueur de tampon possible. Ce paramètre sera le nombre de tampons de cette longueur libérés par seconde.

Les informations dégagées par le simulateur seront le maximum d'octets utilisés lors de la simulation. Cette donnée existera pour chaque longueur de tampon .

Une deuxième notion moins importante sera la place prise en moyenne.

Nous avons également écrit des programmes dont la fonction sera de donner l'état de ces tampons lors d'une exécution réelle.

Ces programmes interviendront dans le calcul des paramètres qui permettront de simuler la libération des tampons.

De plus, ils seront toujours utiles pour le contrôle de l'utilisation de ces tampons par après.

Ces programmes ont été inclus à un outil de mesure existant: RMF, où ils sont devenus une option supplémentaire dans le choix des mesures supportées par RMF.

RMF est un outil de mesure essentiellement orienté vers l'utilisation de ressources. Il permettra de contrôler la validité des paramètres du gérant des ressources (SRM).

C'est pourquoi, nous avons tenu à exposer les grandes lignes de RMF et de SRM.

Enfin, notre méthode sera appliquée sur un cas réel, où nous mettrons donc en pratique les différentes étapes qui nous amèneront à définir les paramètres des tampons.

oooooooooooo

CHAPITRE I: VTAM

Avant de poursuivre l'exposé de ce mémoire, il est nécessaire de décrire le contexte dans lequel on évolue, ce qui permettra de mieux saisir les raisons de certains comportements de VTAM ainsi que les différentes fonctions qui sont à sa charge.

VTAM : Vitual télécommunication access method.

Dans l'exposé de VTAM et du mécanisme interne de son fonctionnement, nous irons en deux étapes: d'abord un bref exposé de VTAM "externe" tel qu'il apparaît à l'utilisateur et ensuite les séquences logiques des opérations d'entrée/sortie en relation avec la gestion de ses tampons: VTAM interne.

I.1 VTAM externe:

VTAM est une tâche du système qui dirige les transmissions de données entre les programmes d'application et les différentes composantes du réseau de télécommunication.

Le principe de base est donc que d'une part, les programmes d'application utilisent VTAM pour communiquer avec les terminaux et que ceux-ci, d'autre part, utilisent VTAM soit pour répondre, soit pour envoyer un message.

Les termes "message" et "transaction" désigneront tout ce qui est susceptible de transiter par VTAM, que ce soient des commandes, des données ou des paramètres de contrôle.

Un programme d'application sera le processus dont la fonction est de traiter les données reçues d'un terminal appartenant au réseau.

I.1.1. Les services pris en charge par VTAM:

Les services pris en charge sont les suivants:

1. établir, contrôler et terminer les accès entre les

terminaux et les programmes d'application.

2. transférer les données entre les terminaux et les programmes d'application.

3. permettre le contrôle et la modification de la structure du réseau.

Conjointement à son rôle primaire dans la transmission des données, VTAM possède des caractéristiques lui permettant d'être la base d'un réseau de grandeur quelconque; ces caractéristiques sont:

- partage des ressources du réseau, ce qui devrait le rendre plus efficace.
- distribution des fonctions de contrôle du réseau, ce qui réduira la charge au niveau processeur central.
- reconfiguration dynamique possible.

I.1.2.-Partage des ressources:

VTAM considère tous les éléments du réseau comme partageables. Les programmes d'application s'exécutant sur le système central peuvent partager les ressources suivantes:

- contrôleur de lignes, les unités de contrôle et les lignes
- les terminaux
- les contrôleurs de communication

I.1.3 - Distribution des fonctions:

Les facilités du contrôleur de communication programmable permettent à VTAM de décharger le processeur central de certaines fonctions de contrôle de lignes, de détection d'erreurs et de leur prise en charge etc...

En général: - VTAM alloue les ressources
- le contrôleur de communication contrôle le flot de données dans le réseau

- des terminaux intelligents formattent leurs données et traitent des transactions locales.

Représentons par un schéma général l'organisation logique du réseau.

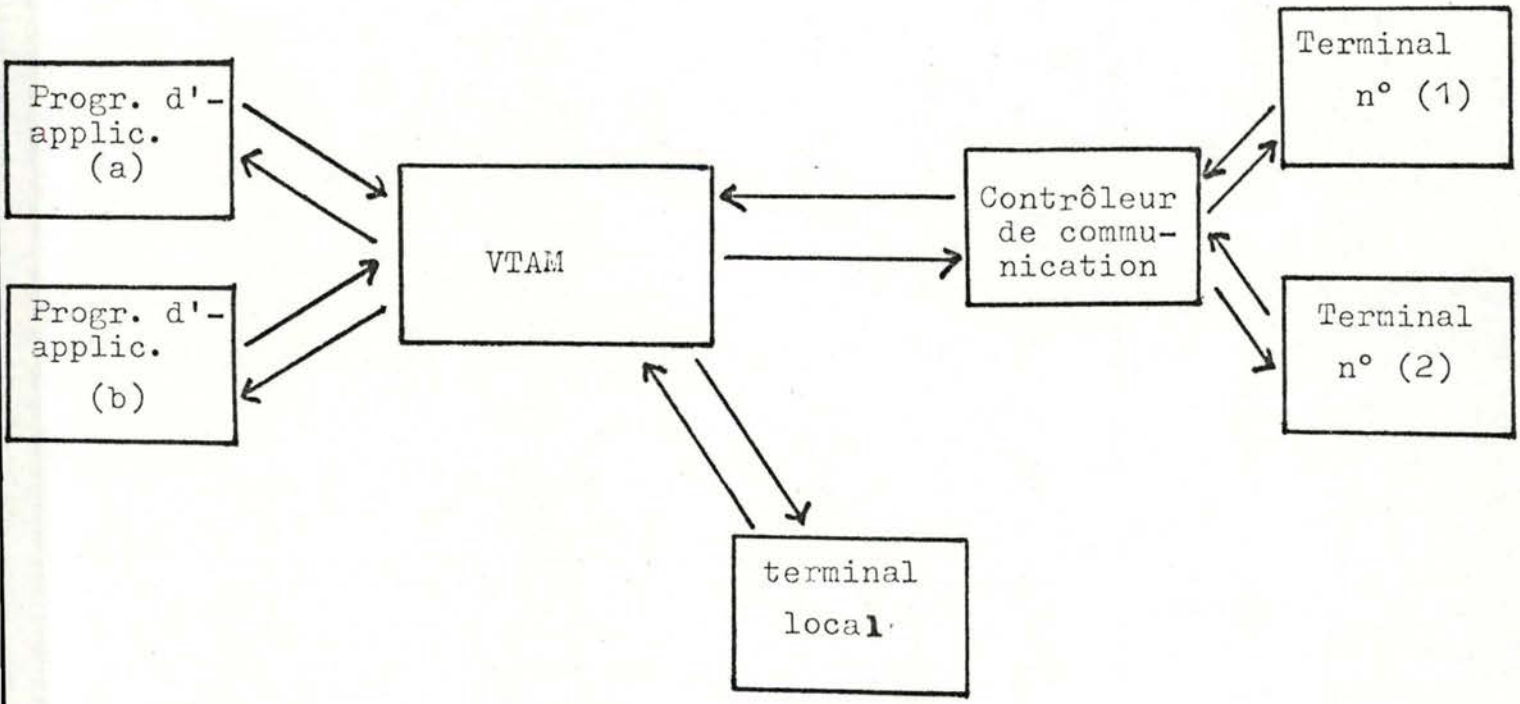


figure I-I

I.1.4: Relations logiques entre les composants du réseau:

Les contrôleurs de ligne, les unités de contrôle et les lignes peuvent être utilisés par plus d'un programme d'application. Un programme d'application ne s'occupe pas de ces contrôleurs de communication, de ces unités de contrôle ni des lignes; il communique uniquement avec un terminal. VTAM contrôle l'utilisation des chemins afin de fournir une connexion logique entre un programme d'application et les

terminaux.

N'importe quel terminal peut communiquer avec n'importe quel programme d'application.

Cependant, un terminal ne peut communiquer qu'avec un programme d'application à la fois.

Sur le schéma I-I, les terminaux (1) et (2) peuvent donc être mis en rapport avec n'importe lequel des deux programmes d'application (a) et (b), mais un ou les deux terminaux ne peuvent être connectés en même temps aux deux applications.

I.1.5. Procédure de connexion:

Les programmes d'application utilisent certaines "macros" de VTAM afin de demander une connexion à un terminal, un transfert de données entre lui-même et un terminal etc...

La connexion est réalisée par un processus qui consiste à rendre un chemin disponible pour la communication entre un programme d'application et un terminal. La connexion s'effectue lorsqu'à une demande, VTAM répond en rendant un chemin disponible.

I.1.6. Procédure de génération:

En fait, dans VTAM, en dehors de la programmation qui utilise les macros prévues à cet effet, la difficulté se situe surtout au niveau de la génération.

En gros, la génération se passe de la façon suivante:

- Définition des programmes de contrôle du réseau qui seront chargés sur le contrôleur de communication.
- Définition des programmes d'application, des terminaux locaux ou à distance, des procédures de connexion (ex. routine de validation d'un "logon")

I.2. VTAM INTERNE : le mécanisme des entrée/sortie en relation avec l'utilisation des tampons.

Passons d'abord rapidement en revue les différentes étapes d'un opération entrée/sortie.

I.2.1. Connexion d'un programme à VTAM:

Un programme d'application se connecte à VTAM au moyen d'une macro qui aura pour effet la mise en place par VTAM d'une structure de blocs de contrôle permettant au programme de demander des services.

I.2.2. Ouverture des sessions:

L'ouverture d'une session entre un programme d'application et une unité logique s'effectue avec une macro. Dans le cas le plus courant, cette demande d'ouverture de session émane d'un utilisateur extérieur qui formule une demande de "Logon". Cette demande est présentée par VTAM au programme concerné en donnant le contrôle à un programme de validation.

Le programme d'application émet alors une macro qui établit entre les blocs de contrôle représentant les deux entités, les liens matérialisant la session. C'est le rôle des services de connexion.

De plus, cette macro provoque le chargement des modules nécessaires à la gestion des échanges qui vont intervenir dans le cadre de cette session.

I.2.3. Echanges de données:

Tout envoi de messages s'accompagne de l'exécution

de trois types de fonction intervenant en séquence.

Ces trois fonctions sont les suivantes;

1/ le programme d'interface

2/ le "control layer" et le TPIOS

Nous allons successivement exposer les fonctions assumées par le programme d'interface puis décrire l'enchaînement chronologique des fonctions prises en charge par le control layer et le TPIOS en entrée et en sortie.

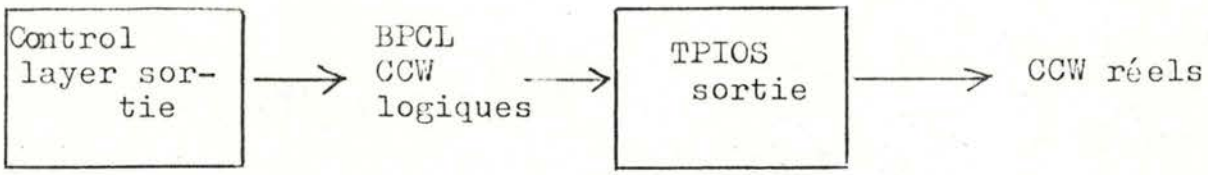
1/ Le programme d'interface:

Ces routines traitent les demandes venant des applications ou des routines VTAM en les contrôlant.

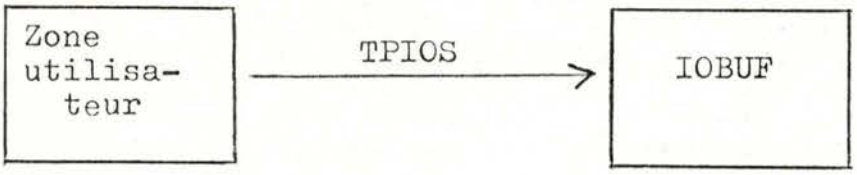
Pour une demande venant d'un programme d'application, ces programmes vérifient la validité de la demande, l'interprètent et, enfin, passent la main au "control layer". Sitôt ces opérations exécutées, les routines avertissent le programme d'application de la fin et fournissent également certaines informations sur les opérations. (code de retour)

2/ Le control layer et le TPIOS en sortie:

Ce schéma représente l'enchaînement de ces deux fonctions avec l'utilisation des tampons.



pour les tampons



a/ le control layer en sortie:

Le control layer exploite le contenu du bloc de contrôle passé par les routines d'interface pour créer un bloc de contrôle appelé "bloc de contrôle canal logique" (BPCL). Il traduit donc sous forme de ccw logiques la demande de l'application. Le BPCL pointe vers la zone à transmettre, située dans l'application. Il passe alors la main au TPIOS en donnant ce BPCL.

b/ Le TPIOS sortie:

Le rôle du TPIOS est de traduire en ccw réels la demande figurant dans le BPCL et de lancer l'exécution du programme canal.

Les données à transmettre référencées par les ccw devant être fixées en mémoire, le TPIOS doit les déplacer au préalable de la zone utilisateurs vers les tampons d'entrée/sortie. Ces tampons sont situés dans les IOBUF.

Cependant, avant d'exécuter le mouvement, le TPIOS

s'assure du fait que rien à priori n'empêchera le message ainsi déplacé d'être transmis vers le noeud suivant. (Nous entendons par noeud, l'étape suivante par laquelle va transiter le message,par exemple, le contrôleur de communication).

Cette précaution, en fait, est prise pour pallier l'engorgement éventuel des "IOBUF" par des messages dont la transmission serait conditionnée par un évènement extérieur.

Si rien n'empêche le mouvement, ce dernier est réalisé.

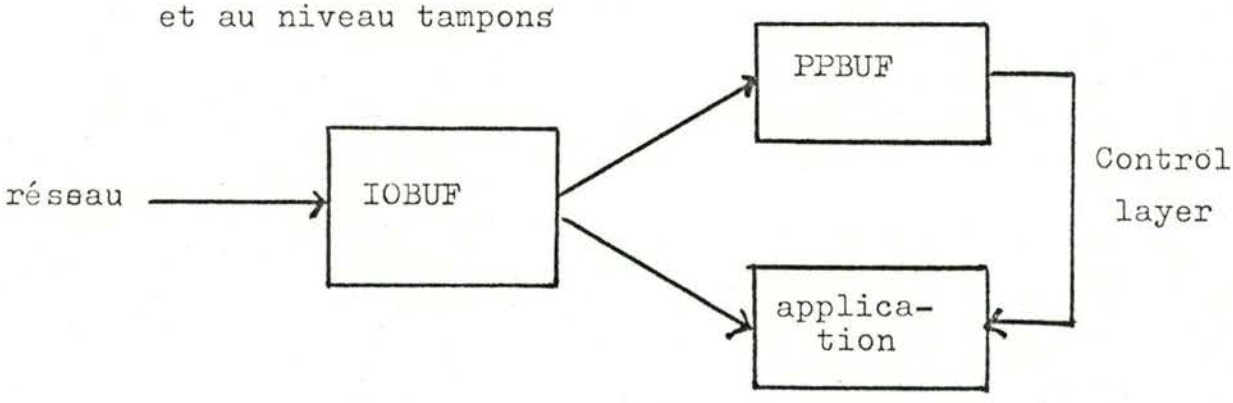
De plus, à chaque opération d'écriture émise vers le contrôleur de communication, le TPIOS chaîne une lecture afin de permettre à celui-ci d'envoyer vers le processeur central des données éventuellement déjà prêtes. Pour cela, le TPIOS réserve un nombre de tampons suffisant (constante définie) pour prendre en charge le message le plus long que peut lui envoyer le contrôleur de communication.

Nous pouvons donc déjà définir les IOBUF: type de tampons destinés à prendre en charge temporairement les messages de sortie et les messages d'entrée comme nous allons le voir.

3/ Le TPIOS et le control layer en entrée :

Représentons également ces fonctions par un schéma

et au niveau tampons



a/ le TPIOS en entrée:

Il va libérer les tampons précédemment réservés et non utilisés. Il passera ensuite la main au control layer d'entrée.

b/ le control layer en entrée:

Celui-ci reçoit des informations composées soit des messages arrivant au réseau, soit des renseignements concernant l'exécution d'une macro émise précédemment.

Dans le premier cas, le control layer libère les IOBUF en réalisant l'une des opérations suivantes:

- transfert de leur contenu vers la zone utilisateur si le processus a émis une demande susceptible d'être satisfaite par le message arrivé.
- transfert du contenu des IOBUF vers les PPBUF. La fonction de ce dernier type de tampons sera de prendre en charge les messages qui ne peuvent être traités immédiatement par l'application (ex. elle est en attente sur une ressource). Ils sont donc destinés à contenir les messages jusqu'à ce que l'application soit prête. A ce moment, le message sera transféré des PPBUF vers la zone utilisateur.

Nous avons décrit le mécanisme général d'une opération d'entrée/sortie ainsi que de la fonction des tampons. Nous allons maintenant décrire les tampons plus en détail.

I.3. LES TYPES DE TAMPONS:

==========

En pratique, il existe onze types de tampons composés entre autres des IOBUF et des PPBUF, décrits précédemment.

Mais la plupart de ces onze types de tampons contiennent des blocs de contrôle relatifs à l'organisation du réseau, au nombre de terminaux.

En d'autres termes, ils sont constants et la définition des paramètres qui les fixeront ne posera guère de problèmes.

Nous allons plus particulièrement nous attacher à deux types de tampons : les IOBUF et les PPBUF.

L'utilisation de ces deux types de tampons dépend de la fréquence d'arrivée des messages ainsi que de la distribution de leur longueur.

Auparavant, définissons les différents paramètres qui concernent un type de tampons et par conséquent les deux types qui nous intéressent.

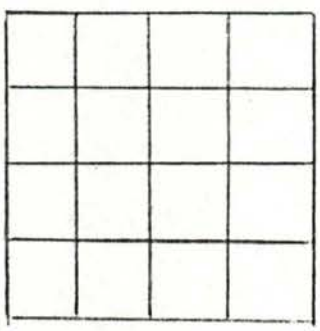
Ces paramètres sont:

- BNO : nombre de tampons
- BTH : limite du nombre de tampons utilisés.
- BSZ : longueur des tampons

I.3.1 : LES IOBUF

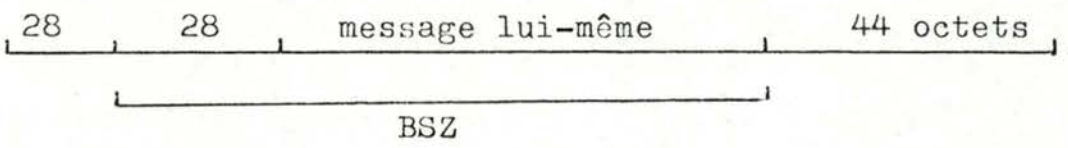
Les deux valeurs BNO et BSZ sont fixes pour toute une session d'activité de VTAM. Ces tampons sont fixés en mémoire principale; ce qui posera le problème du coût de la mémoire. Cette place allouée aux IOBUF devra être suffisante pour prendre en charge les messages venant du réseau avec l'aide des PPBUF.

Si l'on représente par M la place allouée aux IOBUF



M

On a (72 + BSZ) x BNO + 8 = M
où 8 est une constante pour un type de tampon et 72 est la longueur d'une zone constante à chaque tampon IOBUF et utilisée par le système (ex. sert à chaîner tous les tampons que prend un message)



on voit que 28 + 44 = les 72 octets constants
28 est également une zone constante cette fois-ci par message, et le BSZ est celui déclaré à VTAM.
et soit l la longueur du message
x le nombre de tampons employés par ce message.

on a $x = f((l+28) / BSZ)$
ou $f(z) : z \longrightarrow y$ = le plus petit entier $\geqslant z$
et la place totale prise par ce message: x fois (BSZ+72)

Les règles d'allocation sont les suivantes: un seul message à la fois peut occuper un tampon et un message peut occuper plusieurs tampons si la longueur est supérieure à BSZ.

Le premier tampon prendra les (BSZ - 28) premiers octets tandis que les suivants pourront contenir BSZ octets avec éventuellement une perte à la fin du dernier tampon.

I.3.2. LES PPBUF:

Leur BSZ doit être égal au BSZ défini par les IOBUF.

L'importance de la ressource mémoire est moins significative dans ce cas car les tampons sont pageables.

C'est pourquoi leur BNO devra être calculé sur d'autres bases tenant compte également de la différence fondamentale d'utilisation (uniquement messages d'entrée).

La définition des paramètres qui fixeront les IOBUF et les PPBUF ne pose pas de problème tout au moins dans la procédure qui va permettre de les donner à VTAM.

En effet, ils sont situés sur une librairie dont VTAM se sert à chaque fois qu'il est démarré. Nous ne parlions donc pas ici des méthodes de calcul de ces paramètres.

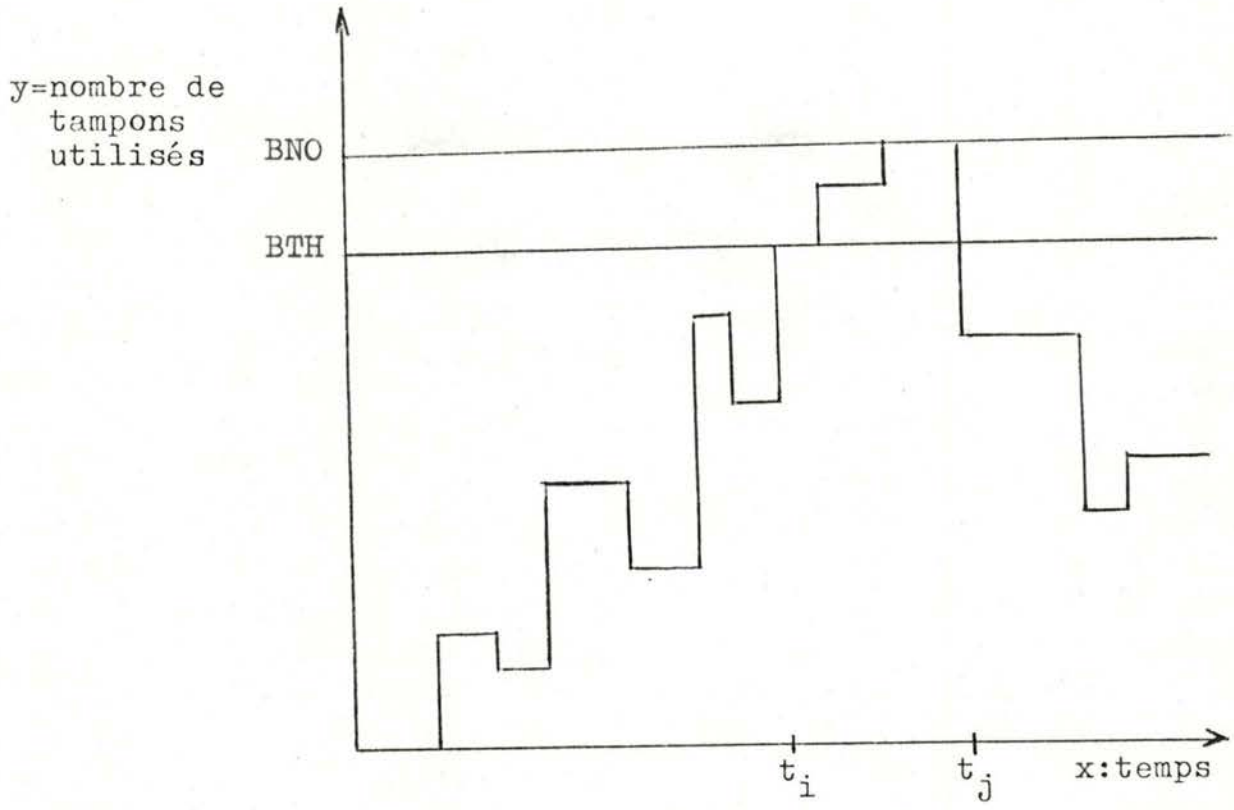
Enfin, les PPBUF possèdent la même structure que les IOBUF et, tout comme ces derniers, répondent à la régle qu'un seul message peut prendre place dans un tampon alors qu'un message peut être éclaté en plusieurs tampons.

I.4: LE RALENTISSEMENT DE VTAM:

I.4.1. Le phénomène en lui-même:

Le phénomène du ralentissement est dû au fait que le nombre d'IOBUF disponible tombe en-dessous d'un niveau défini à l'avance. Ce niveau est le paramètre BNO - BTH où BTH est évidemment le seuil du nombre de tampons utilisés. Ce phénomène ne joue qu'au niveau des IOBUF et pas des PPBUF.

Si l'on représente par un graphique:



Si le nombre de tampons utilisés se situe entre BTH et BNO, soit BTH $\leq y \leq$ BNO, nous nous trouvons dans cette situation. L'intervalle (t_i , t_j) représente la durée pendant laquelle le phénomène persiste.

La réaction de VTAM à un état semblable sera de refuser tous les messages qui ne sont pas des demandes privilégiées. Cette réaction persistera jusqu'à ce que la situation se normalise.

Les demandes privilégiées sont en fait des commandes "expéditives" qui ne souffrent guère de retard dans leur traitement. Elles ont été définies comme étant privilégiées. Elles consistent principalement dans les commandes d'activation et de désactivation.

Ces commandes ont des fréquences extrêmement faibles.

Dans ce cas précis, elles permettront de diminuer la charge du réseau. Ce sera d'ailleurs une mesure extrême.

Les conséquences au niveau du temps de réponse de telles contraintes sont évidemment très graves et devraient être évitées avec le plus grand soin.

I.4.2. Les causes du ralentissement:

Nous allons distinguer deux types de causes qui peuvent entraîner ce genre de situation:

1. celles dues au système.
2. celles dues à l'installation.

I.4.2.1. Les causes système:

Une premier raison système peut être due à une surcharge qui se traduit par une arrivée de messages plus rapide que leur libération. VTAM possède un compteur qui lui dit s'il peut accepter un message. Ce compteur représente le nombre de tampons utilisés. Si celui-ci est inférieur à BTH, le message est accepté.

De manière générale, si BTH n'est pas atteint:

- il accepte les sorties après avoir vérifié la disponibilité du noeud suivant et réserve le nombre de tampons nécessaires.
- il accepte les messages en entrée et réserve auparavant.

Une deuxième raison est due au fait que le terminal ne sait pas se rendre compte de la disponibilité de l'application. Ceci va permettre au terminal en question d'envoyer des messages jusqu'à concurrence d'un nombre maximum de tampons utilisés dans les PPBUF. Ce nombre de tampons maximum permettra de limiter l'envoi de messages au niveau d'un terminal et est une constante propre à chaque terminal définie à VTAM.

Mais si l'importance de ces envois pour tout le réseau a pour conséquence le remplissage complet des PPBUF, les prochains messages resteront dans les IOBUF avec, à brève échéance, les conséquences du ralentissement.

C'est, par ailleurs, un des cas de ralentissement les plus graves car il nécessitera probablement la relance du système.

I.4.2.2. Les causes dues à l'installation:

Les méthodes d'évaluation des différents paramètres régissant les tampons ne sont pas extrêmement fiables.

Ceci nous amène à considérer les causes résultant d'une mauvaise définition des paramètres.

En effet, une mauvaise définition des BSZ, BNO, BTH pour les deux types de tampons se répercute à deux niveaux d'ailleurs extrêmement liés : celui d'une mauvaise gestion de la mémoire et, en relation, celui du problème du ralentissement plus crucial par ailleurs.

a/ mauvaise gestion de la mémoire:

Une mauvaise gestion de la mémoire devra être prise dans le sens place mémoire nécessaire (en octets) excessive pour la situation observée.

En d'autres termes, la situation réelle aurait pu être prise en charge au prix d'exigences mémoire moins importantes.

Cela pourrait résulter de deux raisons:

- Une longueur de tampon trop petite verra les messages éclatés en plusieurs tampons et, par ce fait,

une perte de place due à la zone constante de chacun de ces tampons.

- une longueur de tampon trop grande provoquera une perte sensible de place à la fin de chaque tampon.

Nous allons nous efforcer dans la suite d'apporter une solution à ce problème assez crucial.

b/ mémoire insuffisante:

Le point précédent peut déjà être la cause d'un ralentissement si la perte de place provoque l'atteinte de BTH, alors qu'une définition adéquate eût peut-être permis d'éviter ce problème.

- le paramètre BTH: Pour ce paramètre également, des précautions doivent être prises, car il doit répondre à deux objectifs. Il doit d'abord être suffisamment grand pour ne pas déclencher des ralentissements intempestifs et évitables avec un BTH supérieur.

Ensuite, il doit laisser suffisamment de place au nombre de tampons restant, c'est-à-dire BNO-BTH, de manière à pouvoir traiter les demandes privilégiées en cas de ralentissement.

- le paramètre BNO: ce paramètre doit être suffisamment réaliste au vu des besoins du système et du BSZ choisi et, d'autre part, ne pas être excessif au détriment de la ressource mémoire.

Il s'agit d'adopter une stratégie de juste-milieu en ce qui concerne BTH et BNO.

Dans le cas de BSZ, on choisira une taille correspondant aux besoins réels qui peuvent consister en une brusque poussée du système.

C'est le problème auquel nous allons nous attaquer à

savoir l'estimation des différents paramètres qui caractérisent les deux types de tampons IOBUF et PPBUF. Nous allons auparavant étudier les méthodes existantes ainsi que, immédiatement, les outils dont nous nous servirons dans notre méthode.

I.5. LES OUTILS EXISTANTS:

I.5.1.:Le "suivi"

Une autre caractéristique de VTAM est de pouvoir disposer , grâce à GTF, de renseignements concernant les messages qui sont passés par le système.

C'est le suivi interne de VTAM qui va donner par message et pour tous les messages à GTF:

Par message: - entrée ou sortie
- destination
- origine
- longueur du message
- code du message - commande etc...)

et tous les messages, un enregistrement "time o clock" c'est-à-dire le nombre de micro-secondes depuis le début de ce siècle jusqu'au moment où cet enregistrement "time o clock" a été créé par GTF. Ces données sont disponibles en pratique en vrac par l'outil de mesure GTF qui, en fait, ne fait qu'écrire sur bande ce que lui passe VTAM, sauf le time o clock qu'il crée.

Nous pouvons donc avoir ces bandes à notre disposition.

I.5.2. Les sorties des programmes dans RMF:

RMF est un outil de mesure, à la fonction bien définie, qui fera l'objet d'un chapitre ultérieur ainsi que

les différents programmes que nous avons écrits.

Pour l'instant, exposons simplement la liste des renseignements dont nous disposons.

Pour chaque type de tampon, nous pouvons avoir deux types de renseignements: les premiers seront un résumé sur un intervalle, tandis que les seconds un instantané de la situation des tampons.

Pour la session résumée, nous avons:

Par type de tampon:

1/ longueur du tampon

2/ nombre de tampons

3/ nombre de tampons disponibles: 3 données:minimum
-maximum
-moyenne

4/ nombre maximum de tampons utilisés, également trois données: - minimum
- maximum
- moyenne

Ce genre de renseignements permettra d'avoir une idée de l'activité de l'utilisation des tampons sur un intervalle quelconque (option) grâce à ce résumé.

Pour la session "instantané", nous avons:

Par type de tampons:

1/ longueur du tampon

2/ nombre de tampons

3/ nombre de tampons disponibles au moment de l'observation

4/ nombre maximum de tampons utilisés sur le cycle d'observation.

Ces données sont disponibles au terminal. Il est à remarquer que les deux dernières ne seront pas nécessairement complémentaires par rapport au nombre de tampons total, la troisième étant déjà elle-même un maximum sur un cycle d'observation. Cette session permet une observation détaillée de l'évolution dynamique de la situation des tampons.

0000000000

CHAPITRE II : LES METHODES DE DEFINITION EXISTANTES EXPOSE ET CRITIQUE:

Dans l'étude des méthodes existantes, nous allons successivement aborder le problème de deux points de vue: le point de vue de la définition à priori dans le contexte du début d'une installation et le point de vue à postériori, c'est-à-dire le contrôle et ajustement des différents paramètres.

II.1. Définition à priori:

Nous devons définir les paramètres sans connaître la charge réelle.

Il est donc nécessaire de se baser sur des études prévisionnelles de la charge future.

La méthode proposée par IBM se décompose en deux étapes :

1/ estimation d'une longueur de tampon (commun)

2/ calcul d'un BTH, puis d'un BNO approprié pour les IOBUF, puis d'un BNO pour les PPBUF.

II.1.1. Estimation d'une longueur de tampon.

IBM propose une table de BSZ qui va de 88 à 4096 par pas de 8 dans laquelle devra être choisie la valeur adéquate. L'aide fournie se limite à cela et suppose donc la détermination de ce BSZ être un travail de l'installation.

D'autre part, il est nécessaire d'avoir à sa disposition la distribution de la longueur des messages, ce qui est loin d'être évident.

En effet, si l'analyse des caractéristiques propres à chaque application permettra raisonnablement de connaître le taux des transactions et d'autres données nécessaires à l'implémentation d'une application, la distribution de la longueur des messages ne trouvera peut-être pas sa place dans une telle analyse.

De plus, même si cela était, l'intégration des caractéristiques de chaque application dans un modèle final au vu des interactions, loin d'être négligeables, risque de poser de gros problèmes.

Relevons les difficultés existantes:

1. La plupart des utilisateurs ont certaines difficultés à estimer leur futur comportement.

2. Pour un même terminal, les utilisateurs peuvent être différents et chacun possède sa propre méthode de travail.

3. Le nombre de terminaux différents et les genres d'utilisation augmentent de manière sensible la difficulté de greffer un modèle rigoureux qui respectera le comportement du réseau (TSO, TP, ...).

4. De la même façon que pour le point 1 et cette fois-ci concernant plus particulièrement le BSZ, les utilisateurs n'ont pas beaucoup la notion de la longueur des messages qu'ils auront à envoyer.

5. L'heure a également une importance primordiale dans cette étude, car les intervalles d'observation risquent d'enregistrer des variations importantes.

On se rend compte du nombre des caractéristiques qui interviennent dans le comportement du réseau.

Tous ces différents problèmes rendent extrêmement hasardeux l'adaptation du modèle résultant de cette étude au modèle futur.

Nous avons relevé les principales difficultés concernant une analyse préalable des deux paramètres qui interviendront, à savoir les taux et les longueurs des messages.

Nous reprochons donc à la méthode concernant l'estimation de la longueur du tampon de laisser place à un vide que ne pourrait combler qu'une analyse pour le moins périlleuse.

II.1.2. CALCUL d'un BTH pour les IOBUF:

Une fois la longueur du tampon déterminée, il reste donc à calculer le BTH et le BNO.

Le BTH sera calculé selon la formule suivante:

$$BTH = (MAXBFRU_{cc} \times 0,40) NTRAN_{cc}$$

Nous avons limité volontairement le développement de la formule de calcul au niveau du contrôleur de communication.

En effet, d'une part le principe est le même pour les autres unités et, d'autre part, l'exposé complet de la formule nous aurait amené à entrer dans la description de tous les types de terminaux susceptibles d'exister dans une installation.

Le principe est de pouvoir supporter la majorité des messages venant du contrôleur de communication.

1. MAXBFRU désigne le nombre de tampons que prend le message de longueur maximale passant par le contrôleur de communication.

2. Le facteur $NTRAN_{cc}$ est le taux moyen des transactions passant par le contrôleur de communication. Soulignons à ce propos qu'aucune méthode d'estimation n'est proposée.

3. Le facteur 0,4 est un facteur de proportion qui devrait permettre d'obtenir au total un nombre de tampons capable de prendre en charge tout le flot de messages passant par le contrôleur de communication. Il est donc supposé suffisamment élevé pour supporter les variations dans le taux des transactions.

Cette valeur 0,4 a été calculée sur base d'un réseau regroupant tous les genres d'application possibles avec une répartition respectant les cas généraux.

En d'autres termes, cette formule risque fort de ne pas correspondre à une entreprise particulière car chacune possède des caractéristiques bien spécifiques. Le facteur 0,4 devrait être propre à chaque installation.

II.1.3. Le nombre total de tampons pour le IOBUF

$$BNO_{IOBUF} = BTH + 12$$

De nouveau, nous reprochons à cette formule d'être trop générale et de ne pas refléter une réalité particulière.

De plus, le facteur 12 est fort sujet à caution, car il devrait être suffisant pour traiter les demandes privilégiées; or, cet obectif dépend de la valeur BSZ, qui n'intervient pas dans cette formule.

II.1.4. Le nombre de tampons pour les PPBUF:

$$BNO_{PPBUF} = 2 (t_1 + ... + t_n)$$

où t_i = nombre de terminaux de type i (même longueur maximale de message), connectables au contrôleur

de communication.

Le but est d'allouer pour le PPBUF une place égale au nombre maximum de messages en entrée, susceptibles d'être présents dans le système à tout moment donné.

En pratique, on s'aperçoit que ce type de tampon verra la plupart du temps un nombre relativement important de tampons non utilisés.

La réaction devant ce genre de problème ne devra pas être de réduire le nombre de ce type de tampons.

En effet, d'une part, leur coût est relativement peu élevé et, d'autre part, il sera d'une grande utilité dans le cas où le taux d'arrivée dés transactions sera supérieur au taux auquel les messages seront traités par les processus.

On peut notamment reprocher:

1. le facteur 2 se voudrait être le nombre de messages maximum que peut envoyer un terminal. Or, ce facteur est propre à chaque terminal et fait partie des paramètres du réseau précisés à VTAM.

2. Les différents types de terminaux n'ont pas la même longueur maximale de message.

3. Cette formule ne fait aucune allusion au BSZ, ce qui la rend fort sujette à caution.

II.2. LA PRATIQUE

La difficulté d'appliquer ces formules et les réserves formulées quant à leur validité ont fait qu'en pratique, elles ne sont, dans la plupart des cas, pas appliquées comme nous venons de l'exposer, ce qui montre bien la fragilité de la méthode.

En fait, c'est surtout la détermination de BSZ qui subit des modifications sensibles.

On peut en relever deux:

II.2.1. Estimation comparative du BSZ:

Par rapport à une installation qui a sensiblement les mêmes caractéristiques, on estime le BSZ et puis, ensuite, on procède à l'application des formules.

Mais comme la méthode théorique ne propose aucun moyen de le déterminer, le changement consiste simplement dans le fait que la méthode de détermination ne concorde pas avec la méthode implicitement proposée, à savoir une détermination analytique.

De plus, ce BSZ, malgré la correspondance générale, risque de ne pas répondre totalement aux besoins réels et propres à l'installation.

Enfin, le nombre de tampons obtenus grâce aux formules ne sera souvent qu'une estimation minimale.

II.2.2. Par tâtonnement:

Une méthode également employée est d'essayer une longueur de tampon plus ou moins réaliste et ensuite d'appliquer les formules pour voir ensuite les résultats.

Ces deux méthodes pratiques ont en commun qu'elles fixent un BSZ plus ou moins quelconque et qu'ensuite, elles procèdent à un contrôle et à des changements de valeurs.

On mesure bien les difficultés inhérentes à une telle procédure de contrôle.

En effet, si l'on observe que les tampons sont insuffisants, on peut prendre deux attitudes:

- BSZ:mauvais
- BTH: insuffisant

Si l'on part de la dernière attitude, on risque rapidement d'amener des contraintes mémoires largement excessives.

Si l'on part de la première attitude, on ne sait dans quel sens orienter le BSZ.

De plus, le fait qu'il soit valable peut tout simplement vouloir dire que le BTH s'est avéré suffisant dans la situation observée.

Mais, en fait, ceci constitue déjà les difficultés du contrôle à postériori.

Les moyens qui sont fournis à l'utilisateur font l'objet du paragraphe suivant.

II.3. DEFINITION à postériori:

Nous nous trouvons donc dans le cas où les paramètres sont définis et où nous avons à contrôler l'utilisation des tampons.

A cet effet, VTAM met à notre disposition un outil, le "suivi".

Cet outil ne nous donne que des renseignements par message, ce qui risque de poser quelques problèmes si on veut avoir une idée générale.

Les routines fonctionnant dans le cadre de RMF vont nous donner l'état des tampons IO et PP, ce qui permettra simplement de constater l'inefficience de l'organisation existante.

Cependant, elles ne donneront aucune idée sur la raison, ni sur la méthode à employer pour redéfinir les paramètres concernant les deux types de tampon.

Seul, peut-être, le "suivi" de VTAM, utilisé de manière adéquate, pourrait donner des renseignements. Ce sera un des conceptsde base de notre méthode.

Il n'existe donc pas, à proprement parler, d'outils susceptibles de nous amener à trouver des BSZ, BTH, BNO répondant aux besoins réels, lacune que nous nous efforcerons de combler.

II.4. Un problème de performance relatif au BSZ:

Il existe une contrainte de performance au niveau de la valeur de BSZ, contrainte généralement admise.

En effet, cette contrainte n'est pas indispensable.

Le BSZ doit être au moins égal à la moitié du plus long message qui peut être envoyé par une bonne partie des terminaux.

Autrement dit, cela veut dire qu'on ne doit prendre que deux tampons au maximum pour la plupart des messages venant de la part de ces terminaux (en général, car cela dépend de la configuration, elle est de l'ordre de plus de cent octets). Il s'agit bien de la plupart des messages, car il est évident qu'on ne peut tenir compte des gros envois tels que l'envoi de tout un écran.

Il est clair que la contrainte du nombre des tampons doit intervenir sous une forme ou sous une autre. Mais dire que celle qui vient d'être énoncée répond aux besoins nous semble aller un peu vite.

En effet, d'une part les terminaux concernés enverront rarement un message de longueur maximale, ce qui va défavoriser le point de vue mémoire et, d'autre part, la restriction de deux ou trois tampons par message nous semble limitée.

L'idéal devrait être l'étude d'un rapport coût mémoire sur coût nombre de tampons, qui nous donnerait un BSZ respectant le côté mémoire sans être une source de surcharge pour le nombre de tampons.

L'estimation respective des coûts et surtout leur intégration dans une seule fonction économique est loin d'être aisée, car les deux objectifs sont, à première vue, contradictoires.

Nous disons bien à première vue, car nous verrons que l'objectif mémoire rencontre les contraintes tampons de manière adéquate.

Ce qui veut dire que la contrainte actuellement de mise pourra être évitée dans le cadre de notre méthode qui va donc consister dans un outil d'aide à la définition des différents paramètres.

OOOOOOOOOOO

CHAPITRE III : PROPOSITION D'UNE METHODE

III.1. INTRODUCTION.

Nous nous trouvons devant un problème qui consiste à déterminer des paramètres régissant les tampons de type IO et PP. Ces paramètres sont, rappelons-le, pour les IOBUF (BSZ, BTH, BNO) et pour les PPBUF (BNO).

Nous avons souligné précédemment les difficultés existantes ainsi que la fragilité des méthodes proposées.

Ceci nous a amené à élaborer une autre méthode qui tentera de déterminer la taille mémoire nécessaire ainsi que le BSZ adéquat répondant aux besoins de l'installation.

Ceci concerne les IOBUF. Les PPBUF feront l'objet d'une démarche ultérieure.

Les objectifs seront:

1/ Meilleure utilisation en moyenne de la mémoire.

2/ Eviter la ralentissement.

3/ Pouvoir traiter les demandes privilégiées en cas de ralentissement.

Ces trois objectifs seront atteints grâce à :

1. L'application d'une formule donnera, pour chaque BSZ, la place prise en moyenne. Cette valeur entrera en ligne de compte ultérieurement. Cette approche est purement statique (phase 4) . Les références à des phases concernent l'ordinogramme qui suivra.

2. Nous limiter à prendre le BSZ qui possède cette valeur minimale peut créer certains problèmes. C'est pourquoi nous avons créé un simulateur d'allocation et de libération des tampons.

Il nous permettra de dégager les contraintes dues au ralentissement.

Ces contraintes se traduiront sous deux formes:

- les BSZ ne correspondant pas aux besoins seront détectés et refusés. En d'autres termes, ils n'interviendront pas dans le choix final.
- la taille mémoire nécessaire et suffisante sera dégagée grâce à la donnée MAX_{x_i} (x_i : BSZ possible). Cette donnée MAX_{x_i} représentera le maximum en octets de la place utilisée pour chaque x_i. La taille mémoire nécessaire et suffisante pour la situation observée sera le premier nombre de pages $\geq$ min (MAX_{x_i}).

Les fonctions du simulateur (phase 4) seront globalement les suivantes:

- l'allocation sera aisée grâce à la connaissance de la longueur du message.
- la libération se fera grâce à la connaissance pour chaque longueur de tampon possible (x_i) du nombre de tampons de cette longueur libéré par seconde (P_{x_i}) . Cette valeur sera déterminée auparavant.

Connaissant également les temps d'arrivée de chaque message, il sera possible de simuler la libération des tampons.

Les P_{x_i} seront calculés de la façon suivante (phase 2)

- A l'arrivée de chaque message dans le système, nous imprimerons le temps et le nombre de tampons pris par les messages arrivés jusque là pour chaque longueur possible de tampon.
- Nous déterminerons ensuite la suite des messages libérés.

Les sorties du point précédent permettront la détermination des P_{x_i} grâce à la connaissance du nombre de tampons pris par ces messages libérés et des moments de début et de fin où ces messages ont été libérés.

Enfin, lors de l'utilisation, nous ferons varier la taille de mémoire allouable afin de connaître les contraintes correspondantes à chacune de ces tailles.

3. Nous nous efforcerons de déterminer un nombre de tampons suffisant pour traiter les demandes privilégiées. Ce nombre sera indépendant de la taille mémoire allouée aux tampons. (phase 3)

III.1.1. Problème de définition à priori:

La méthode que nous allons proposer était essentiellement un moyen de contrôle et de décision sur un

environnement déjà existant, nous voyons deux moyens de l'appliquer:

III.1.1.1. Simulation

Grâce à des paramètres déterminés à l'avance, il s'agirait de simuler la situation des tampons IO sur base de messages dont la longueur et le taux sont générés suivant des lois précisées au simulateur IBM.

On peut soumettre ce moyen aux critiques préalablement exposées, à savoir la valeur des paramètres ainsi estimés . Ceci n'empêcherait pas, de toute manière, de calculer les véritables valeurs une fois la situation réelle existante.

III.1.1.2. Utilisation au départ de l'ouil de contrôle et de décision.

En définissant une longueur de tampon réaliste, et en réservant un nombre de tampons volontairement exagéré, nous pourrons arriver aux mêmes résultats au prix d'efforts moindres.

En effet, chaque méthode de définition à priori nécessitera quand même des contrôles tandis que celle-ci ne demande aucune étude préalable.

Le nombre de tampons volontairement exagéré est dû au fait que l'on veut éviter le ralentissement. Il suffira alors de lancer les outils que nous allons définir pour avoir les paramètres désirés.

Nous allons maintenant exposer notre méthode de détermination des paramètres des IOBUF. Il s'agit donc de la méthode à postériori s'adaptant à un environnement existant. Nous aborderons alors finalement le problème du BNO des PPBUF.

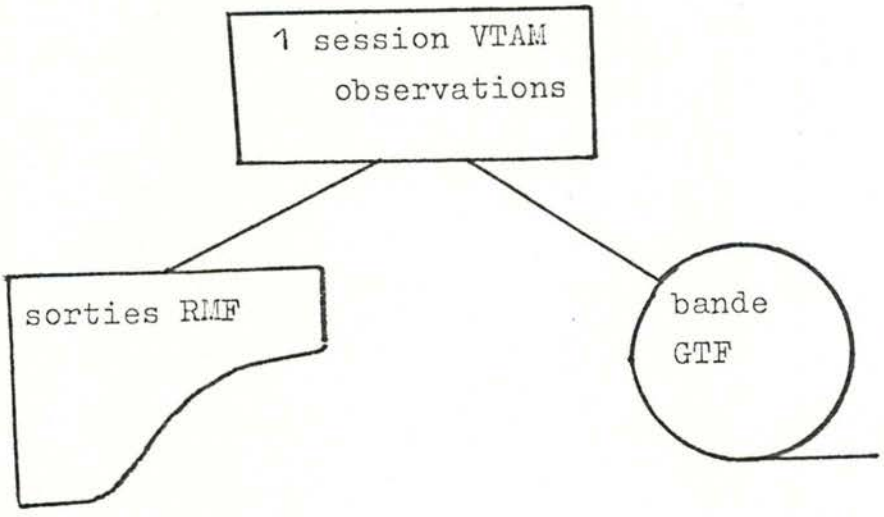
III.1.2. SCHEMA Général:

Comme nous l'avons dit précédemment, notre méthode se basera essentiellement sur les faits observés par lesquels nous entendons l'emploi des données fournies par le suivi de VTAM et les sorties de RMF.

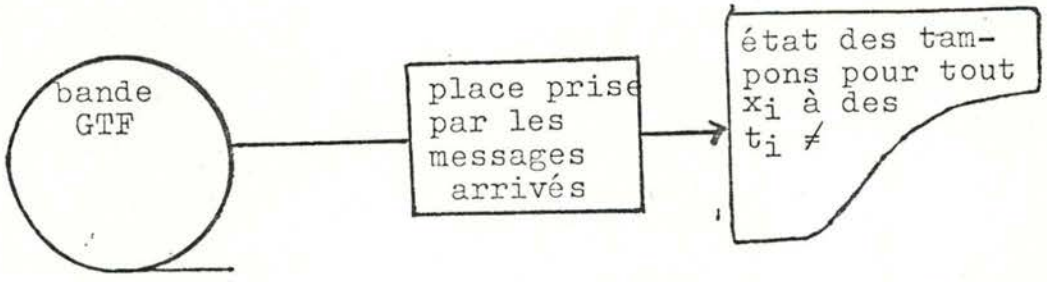
Voici le schéma général de la méthode au point de vue chronologique.

Rappelons que ceci concerne uniquement les IOBUF.

phase 1

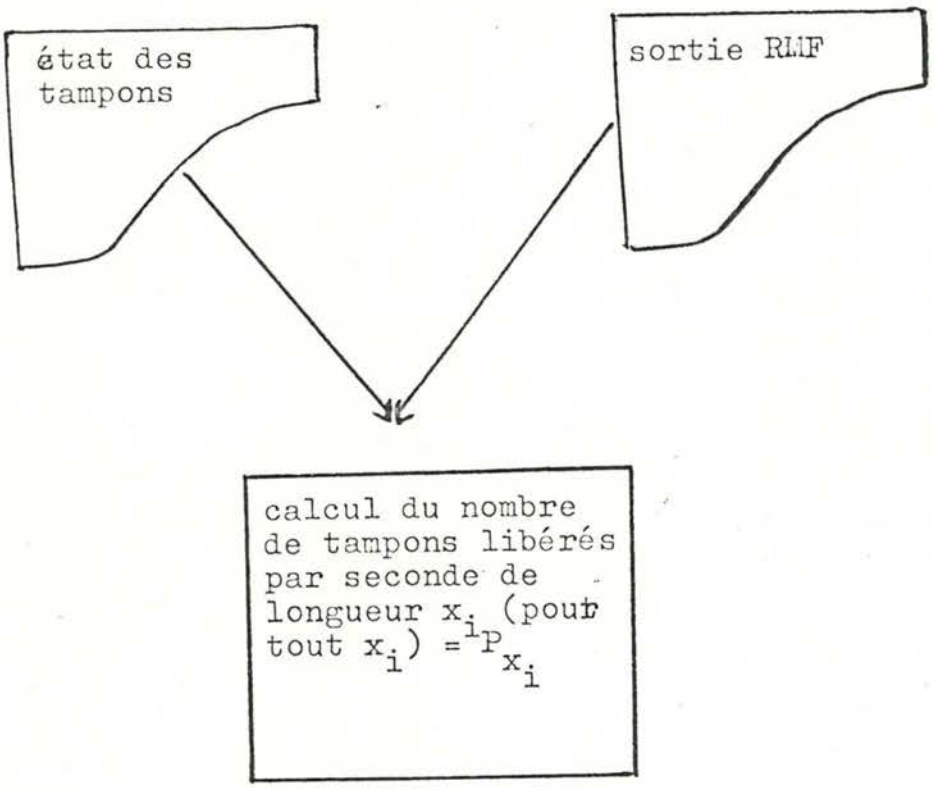


phase 2

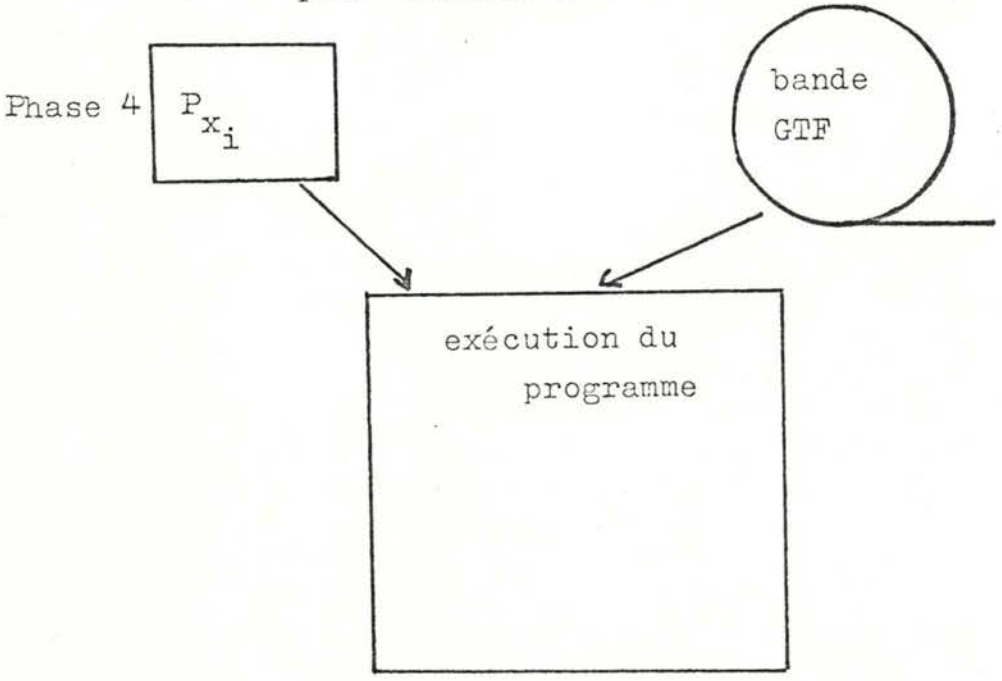


x_i représente les valeurs de BSZ possibles

et t_i représente le fait que l'on a la situation à différents moments .



phase 3: détermination d'un nombre de tampons nécessaires pour traiter toutes les demandes privilégiées



cfr: ordinogramme

ORDINOGRAMME DE

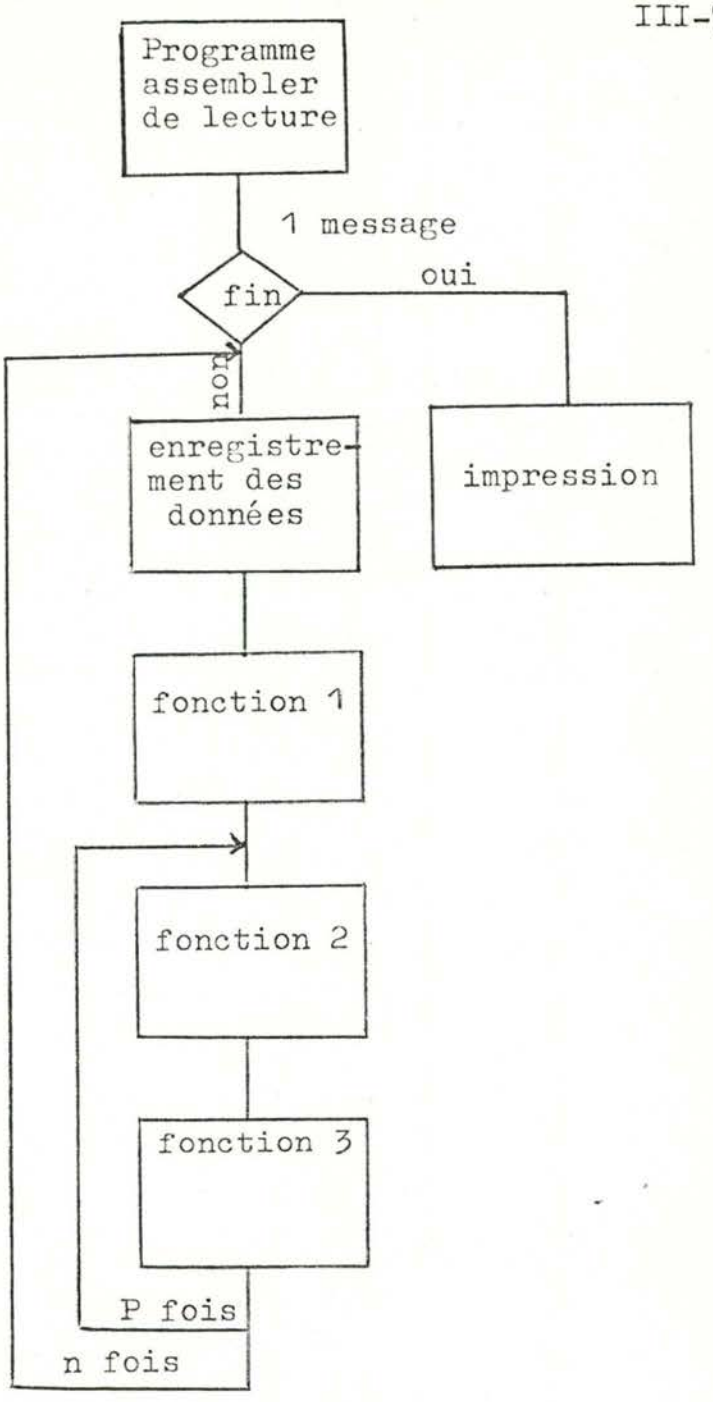
notre programme.

tion 1:
onction 1 consiste dans la
lation de la situation des
ons si le véritable BSZ
t été $x_1 \ldots x_n$.
a donc n valeur de BSZ
ibles.

tion 2:
onction 2 va permettre de
ir si le x_i est valable
la mémoire allouée.
atteint)
x_i

tion 3:
va alors faire varier la
ire allouée et retourner
fonction précédente.
supposons qu'il y a
ailles de mémoire possibles.

nregistrement des données
ortera notamment les opé-
ions de cumul (nombre de
sages, place totale
se, ...)



Dans l'exposé de notre outil, nous ne suivrons pas l'ordre chronologique.

Nous supposerons certaines données introduites quand c'est nécessaire et nous le préciserons lorsque c'est le cas.

En pratique, nous allons procéder comme suit:

Sur une session:

1. exposé des données à notre disposition (phase 1).
2. exposé de la formule de calcul de la place prise par les messages arrivés.
3. exposé de la méthode de simulation en supposant P_{x_i} connus.
4. Nous allons faire varier cette taille afin d'observer les conséquences. (phase 4, fonction 3)
5. Calcul des P_{x_i} (phase 2)
6. Calcul d'un $(BNO - BTH)_{x_i}$ (phase 3)

III.2. Les données disponibles:

Nous disposons de la bande de sortie "suivi VTAM" décrite précédemment. Cette bande nous donne les messages qui sont passés par le système.

Nous disposons également des sorties RMF qui nous donneront la situation des tampons pour la longueur existante dans l'installation à différents moments.

III.3. Détermination d'une longueur de tampon sur une session.

III.3.1. Approche statique:

En premier lieu, nous avons écrit un programme assembler qui se charge de lire la bande et de passer les renseignements nécessaires à un programme COBOL, certains traitements au niveau de l'octet étant nécessaires.

Ces renseignements sont repris dans le contenu du "suivi" GTF.

A partir de la bande GTF, nous allons calculer la distribution exacte de la longueur des messages sur la session observée. Une session est ici une période d'observation.

Nous allons sortir sur toute la session la distribution observée de la longueur des messages suivant cette représentation.

Soit l_1, ... l_n les différentes longueurs de messages possibles.

On aura $\forall\ l_i$

l_i , n_{l_i} , t_{l_i} , f_{l_i}

où n_{l_i} est le nombre de messages de longueur l_i

t_{l_i} est le taux de messages de longueur l_i , c'est-à-dire n_{l_i}/N où N est le nombre total de messages sur toute la session
et est égal à P (un message soit de la longueur l_i) =

t_{l_i} et enfin f_{l_i} la fréquence d'arrivée par seconde qui sera égal à n_{l_i}/S où S est le nombre de secondes de

la session.

Ces sorties ne seront que des renseignements .

Et nous sortirons pour chaque BSZ la valeur calculée de la manière suivante :

(3)

$$\left(\sum_{l_i}^{l_n} \underbrace{n_{l_i} \; f((l_i + 28)\, x_j)}_{(1)} \; \underbrace{(x_j + 72)}_{(2)} \right) (t_n - t_i)$$

où x_j représente les valeurs de BSZ possible.

(1) représente le nombre de tampons pris de longueur x_j par les messages de longueur l_i et de nombre n_{l_i} .

la fonction f : $z \rightarrow y$: le plus petit entier $\geqslant z$.

(2) représente la place réelle prise par un tampon de longueur x_j

dont (1) * (2) représentera la place totale en octets prise par les messages de longueur l_i .

Si l'on fait la somme depuis $l_1 \ldots l_n$, on va donc avoir la place totale prise par tous les messages c'est-à-dire la place que prendraient les messages s'ils étaient mis les uns à la suite des autres.

A la fin de la session, nous sortirons cette valeur pour tous les BSZ possibles ainsi que la moyenne des octets prise par seconde.

Cette moyenne sera égale à la valeur de l'expression (3) divisée par $(t_n - t_1)$ qui est la durée de l'intervalle.

Si l'on représente sur un schéma la signification de cette donnée:

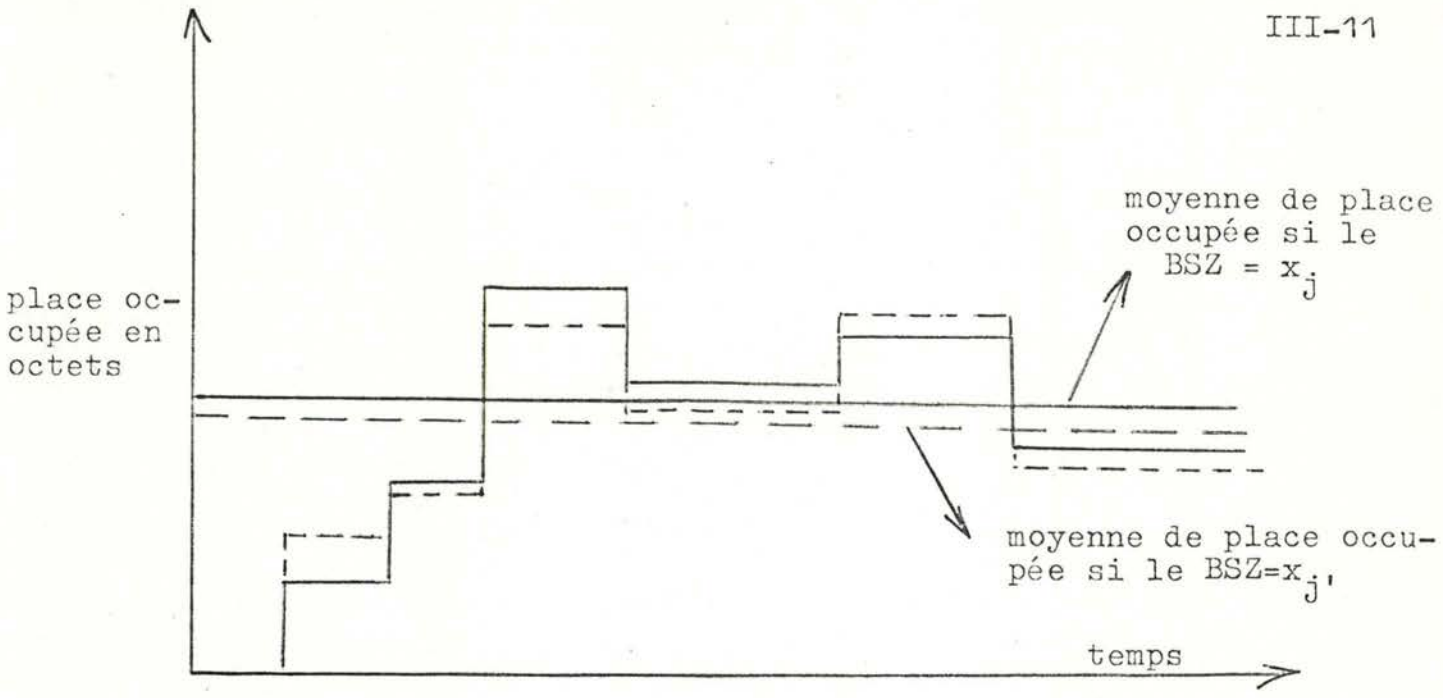


Fig. II-II

C'est-à-dire que $BSZ_{x_{j'}}$ ainsi déterminé va nous donner la meilleure utilisation de la mémoire en moyenne.

III.3.2. CRITIQUE:

A ce stade, nous pourrions déterminer un BSZ adéquat pour un modèle où les fréquences réelles d'arrivée de chaque longueur de message seraient constantes .

Ce BSZ serait égal au x_i pour lequel l'utilisation moyenne de la mémoire serait minimum.

En pratique, il n'en sera probablement rien et l'on pourra observer certaines variations, sources éventuelles de nombreux problèmes.

En effet, une variation importante dans la longueur des messages rendra le BSZ général inadéquat dans ce cas particulier.

De plus, cette méthode ne nous donnera aucune indication permettant de déterminer le BNO.

La variation pourrait provoquer un ralentissement en cas de fréquence d'arrivée suffisamment élevée. La mauvaise correspondance entre la longueur des messages à ce moment et celle de BSZ général prendrait en effet une place prépondérante.

III.4. Construction d'un simulateur d'allocation et de libération des tampons.

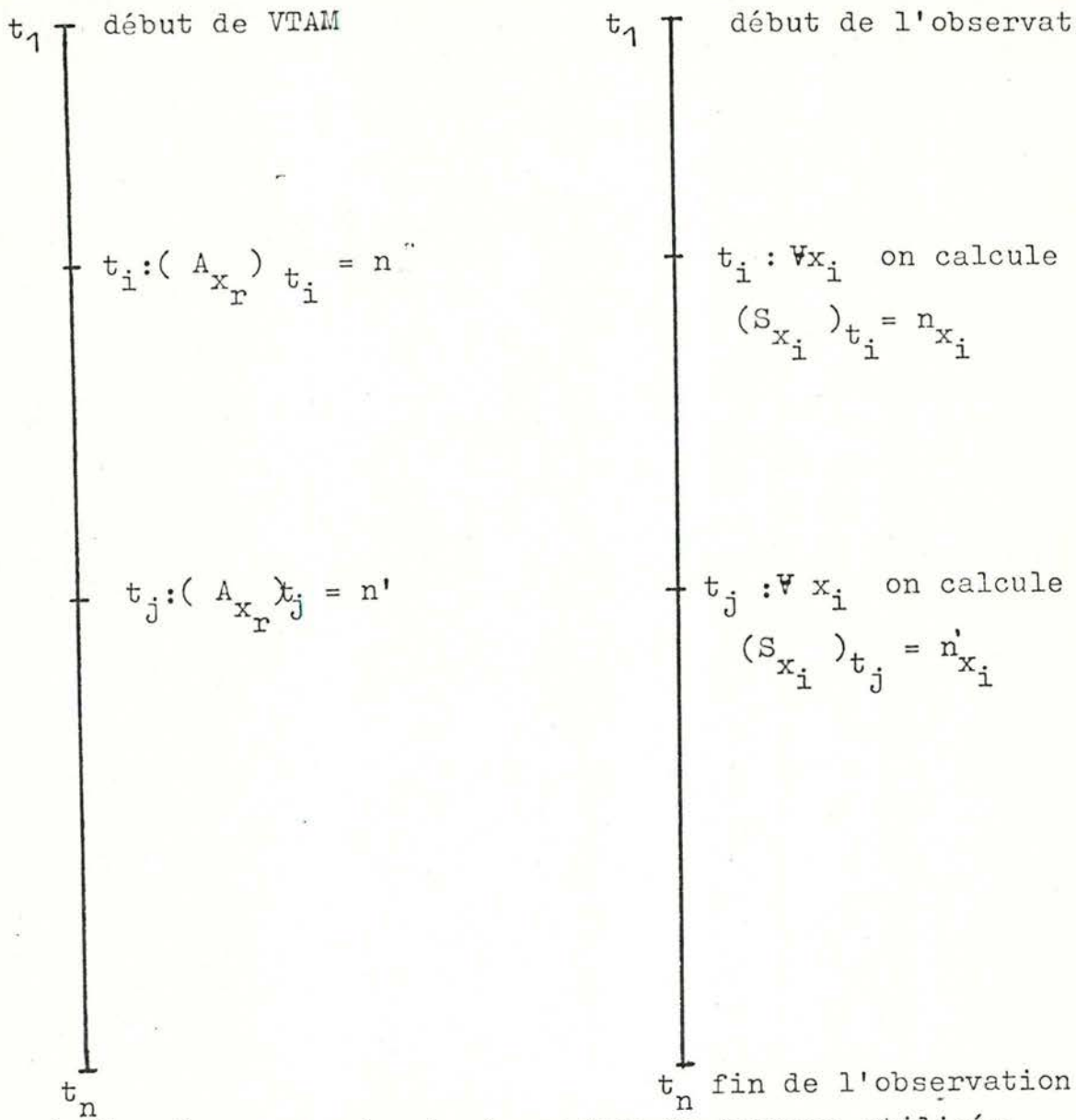
Afin de résoudre le problème soulevé précédemment, nous allons procéder en deux étapes:

1/ estimation, par BSZ possible, d'un nombre de tampons libérés par seconde.

2/ à partir de là, et connaissant les moments d'arrivée des messages dans le système, simulation de ce qu'aurait été l'état des tampons pour chaque BSZ possible, ce qui nous permettra de dégager les contraintes nécessaires.

L'étape de l'estimation d'un nombre de tampons libérés par seconde se fera lors d'un développement ultérieur. Nous supposons donc que ces données sont introduites dans le programme.

Si l'on représente par un schéma les fonctions assumées par ce simulateur:



où $(A_{x_r})_{t_i}$ représente le nombre de tampons utilisés réellement de longueur x_r qui est le BSZ de l'installation, et où $(S_{x_i})\ t_i$ représente le nombre de tampons de longueur x_i utilisés à l'instant t_i, si leur longueur dans le BSZ de l'installation avait été x_i . On va calculer cette valeur pour chaque t_i où t_i représente le moment d'arrivée d'un message et pour tous les x_i possibles.

Nous allons maintenant exposer la façon de calculer le (S_{x_i}) à un instant t.

Soit t l'arrivée du message.

Soit $(S_{x_i})_{t-1}$ la situation après l'arrivée du précédent message.

Rappelons que - nous connaissons pour chaque message son moment d'arrivée ainsi que sa longueur,

- nous connaissons pour chaque x_i le taux de tampons libéré par seconde = P_{x_i}

et on a :

$$(S_{x_i})_t = (S_{x_i})_{t-1} + b_{x_i} - P_{x_1} \times T \quad (a)$$

où b_{x_i} représente le nombre de tampons de longueur x_i pris par le message arrivant

où T est le temps séparant l'arrivée du message de l'arrivée précédente.

On va alors faire le test:

$$(S_{x_i})_t < BTH_{x_i} \quad (b),$$

et BTH_{x_i} représentant le seuil du nombre de tampons de longueur x_i .

On va également calculer Max_{x_i} qui représentera le maximum de tampons utilisés de longueur x_i sur toute la session.

Si le test (b) est négatif, cela voudra dire que l'on a atteint le seuil pour cet x_i. Cet x_i aurait donc provoqué le ralentissement.

La réaction consistera à rejeter cet x_i .

Pour que x_i soit une valeur possible, il faut qu'à tout instant de la session, la relation $(S_{x_i})_t < BTH_{x_i}$

soit vérifiée.

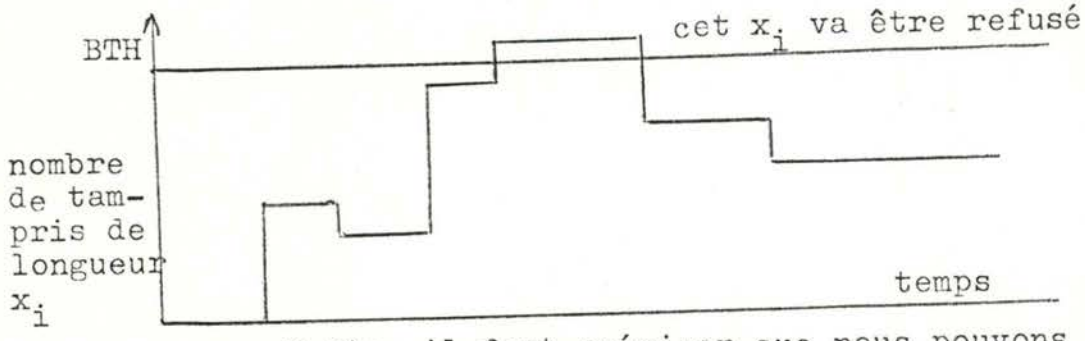
Pour chaque message arrivant dans le système, et pour chaque x_i, nous allons calculer la valeur (S_{x_i}) et faire le test (b).

Si bien qu'à la fin de la session, nous aurons:

Soit C = l'ensemble des x_i possibles.

$$C = \left\{ U\,(x_i) \mid (S_{x_i})_{t_j} < ((BTH)_{x_i} \text{ pour tout } t_j \right\}$$

Si l'on représente par un dessin l'introduction des nouvelles contraintes:



Enfin, il faut préciser que nous pouvons sortir les mêmes données sur des intervalles de longueurs variables (l'option étant le nombre de messages)

Ces données sont donc: $\forall_{x_i}$ Max_{x_i}

(S_{x_i}) au début de l'intervalle

(Max_{x_i}) $(x_i + 72)$ la place maximum prise

$E\,(S_{x_i})$ le nombre moyen de tampons employés de longueur x_i

$(E\,(S_{x_i}))$ $(x_i + 72)$ la place moyenne prise en octets.

le nombre de pages pour lequel cet x_i est possible.

Toutes ces données sont reprises, bien sûr sur l'ensemble de la session.

III.5. Simulation sur différentes tailles de mémoire allouable.

Supposer à priori que la place allouée aux IOBUF est fixée, limite assez fortement l'intérêt de la méthode.

C'est pourquoi nous avons inclus la simulation sur des tailles d'IOBUF allant de 1 à N pages (constituant des limites pratiques).

Le découpage en page est arbitraire mais nous pensons que ce partage est suffisamment affiné pour ne pas créer des problèmes de place perdue et, d'autre part, il reflète fortement la situation réelle.

Soit P_i un nombre de pages possibles

le test lors de la session $(S_{x_i}) \prec BTH_{x_i}$

va devenir

$(S_{x_i}) \prec (BTH_{x_i})_{P_i}$

Les formules définies en III.4 vont prendre leur forme définitive sur une session

$\forall P_i$

C deviendra dans la formule

$$C_{P_i} = \{ U(x_i) \ (S_{x_i}) \ t_j \prec ((BTH)_{x_i})_{P_i} \text{ pour tout } t_j \}$$

Pour chaque P_i , nous aurons les contraintes qui s'y adaptent et elles seront différentes d'un P_i à un autre.

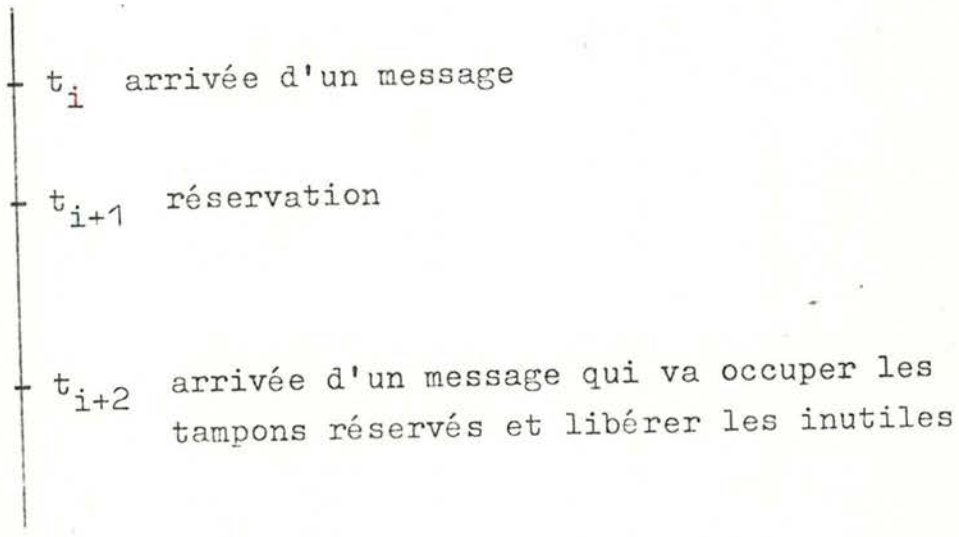
En effet, si les P_i augmentent, les contraintes vont s'élargir, ce qui pourra donner des BSZ différents de ceux trouvés dans le P_i précédent.

En d'autres termes, nous avons actuellement pour chaque P_i l'ensemble des x_i possibles.

REMARQUE: le problème de la réservation.

C'est un problème qu'il nous est impossible d'implemanter dans l'état actuel des outils du suivi car la réservation s'effectue en dehors de l'arrivée des messages dans le système suivant un mécanisme que nous ne saurions simuler.

Si l'on représente ce phénomène



La conséquence pratique d'un tel mécanisme est que le nombre de tampons disponibles peut varier d'une constante maximale = $(MAXBFRU)_{x_i}$ entre deux messages.

Par contre, la situation simulée par nos programmes devra être équivalente à l'instant t_{i+2} .

Ce qui nous amène à dire que le risque se situe dans la possibilité que la réservation provoque un ralentissement.

D'autre part, ce ralentissement non détectable ne durera que le temps t_{i+1} à t_{i+2} , temps extrêmement court, la réservation et l'occupation des tampons constituant des opérations presque simultanées .

On voit que le risque est extrêmement faible et que, de plus, les conséquences sont fort réduites, sinon insignifiantes.

III.6. Calcul pour chaque BSZ du taux libéré par seconde (phase 2)

On a supposé précédemment ces différentes valeurs introduites dans nos programmes. Voyons maintenant la méthode employée pour l'obtention de ces paramètres existant pour chaque x_i disponible.

III.6.1. Outils.

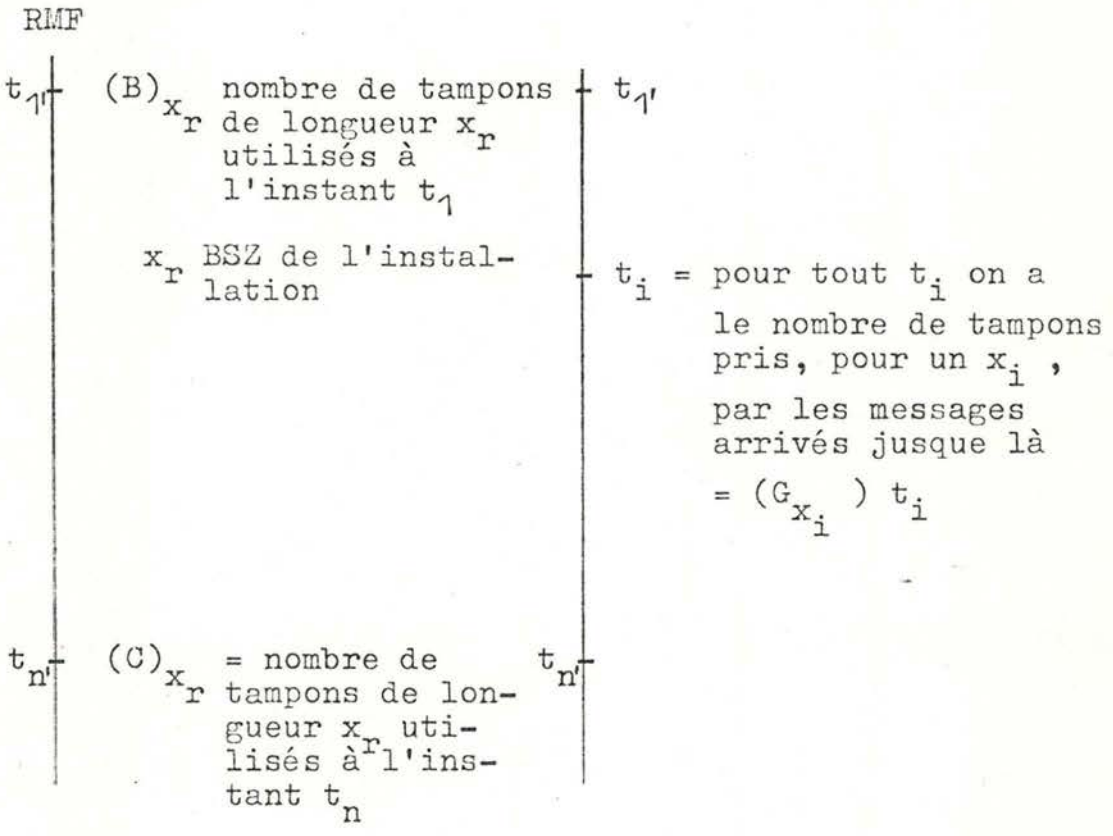
Nous allons à cette fin nous servir de deux outils: d'une part de l'observation faite par nos programmes tournant dans le cadre de RMF et,d'autre part,de nos programmes de simulation.

Nous allons paramètrer de manière:

1. à réduire l'intervalle à un message (option), ceci afin d'avoir le temps d'arrivée de chaque message car les temps n'étaient imprimés, avant, que pour le début d'intervalle.
2. Nous introduisons des $P_{x_i} = 0$. Si on regarde la formule de a/ du § III.4, on observera que, de

cette manière, les $(S_{x_i})_t$ seront simplement la place totale prise, jusqu'à l'instant t. La réduction de la taille d'un intervalle à un message permettra de connaître le nombre de tampons pris par les messages arrivés jusque là, et ceci pour chaque longueur de tampon, puisqu'on imprime la situation des tampons à chaque arrivée de message.

En pratique:



III.6.2. Nombre de tampons libérés de longueur x_r

Au préalable, nous synchroniserons les deux observations en choisissant deux instants $t_{1'}$ et $t_{n'}$ communs aux deux types d'observation:

$$l_{x_r} = (B)_{x_r} + (G_{x_r})_{t_{n'}} - (C)_{x_r}$$

où l_{x_r} est le nombre de tampons de longueur x_r libérés dans l'intervalle $(t_{1'}, t_{n'})$

et t_r est le moment t_i où $(G_{x_r})_{t_i} = l_{x_r}$

c'est-à-dire que t_r est le moment auquel tous les messages arrivés seront libérés.

III.6.3. Nombre de tampons libérés $\forall_{x_i}$ (= P_{x_i})

Ayant déterminé cet instant t_r , nous connaissons donc le nombre de tampons pris par les messages arrivés pour tous les x_i .

on aura donc $P_{x_i} = (G_{x_i})_{t_r} / (t_{n'} - t_{1'})$

qui sera égal au nombre de tampons de longueur x_i libérés par seconde sur l'intervalle.

Il est à remarquer que tous les calculs décrits dans ces paragraphes ont été faits à la main, car, d'une part, nous ne pouvions disposer dans un même programme des deux genres de données et, d'autre part, les calculs sont extrêmement simples et fort réduits.

De plus, quand nous avons parlé de paramètrer nos programmes, nous entendions une recompilation avec toutefois des changements extrêmement limités et très simples à introduire. Ceci a été dû à des fins de contraintes de tests.

Nous proposons d'ailleurs une généralisation des programmes par l'introduction dynamique des P_{x_i} , de la taille des intervalles et éventuellement des tailles mémoires différentes sur lesquelles doivent

s'effectuer les tests.

III.6.4. Choix de $t_{n'}$ et $t_{1'}$

$t_{n'}$ sera le moment t_i où on observe le nombre maximum de tampons utilisés de l'installation (sortie RMF).

Ce choix est destiné à rendre $(MAX)_{x_r}$, lors de la simulation, égal au nombre maximum réellement utilisé à l'instant $t_{n'}$.

$t_{1'}$ sera un moment où le nombre de tampons utilisés est suffisamment bas.

En effet, la distribution de la longueur des messages qui occupent les tampons à $t_{1'}$ ne sera pas nécessairement la même que la distribution de la longueur des messages entre les instants $t_{1'}$ et t_r.

Prendre un tel $t_{1'}$ suffira à minimiser ce risque, les différentes possibles étant d'autant plus petites.

Il est possible que certains autres moments dangereux soient détectés.

Dès lors, on appliquerait à nouveau la méthode pour chacun de ces $t_{n'}$.

Chacune de ces exécutions serait considérée comme une session d'observation à part entière (cfr intégration des sessions).

En effet, la distribution de la longueur n'est pas à priori la même sur toute la session.

III.6.5. Profil général

A partir des résultats de la simulation, on pourra avoir une idée générale du profil d'utilisation des tampons. Le nombre de messages par intervalle définira la précision désirée.

Ce profil définira la façon dont l'état des tampons évolue dynamiquement. Il sera calculé pour les x_i possibles lors du décompte final.

Ce concept interviendra ultérieurement.

III.7. Estimation de la place nécessaire aux demandes privilégiées.

Lors d'un ralentissement, il faudra ménager suffisamment de place pour traiter les demandes privilégiées.

Nous allons aborder ce problème sous un angle propre aux demandes privilégiées.

En d'autre termes, notre politique sera de réserver d'office, quel que soit le BNO, un nombre de tampons constant.

Le problème de la détermination du BTH se transformera en celui de la détermination du BNO, puisque le BTH sera calculé en soustrayant de BNO le nombre que nous allons fixer.

Rappelons les caractéristiques des demandes privilégiées:

- elles sont d'une fréquence extrêmement faible.
- elles ne peuvent supporter un retard important.
- elles auront une réponse

$$(BNO - BTH)_{x_i} = (MAXBFRU)_{x_i} - 1 + (PR)_{x_i} + (RE)_{x_i}$$

où PR est le nombre de tampons de longueur x_i nécessaire pour prendre en charge la demande privilégiée la plus longue
et RE est le nombre de tampons de longueur x_i nécessaire pour prendre en charge la réponse la plus longue.

$(MAXBFRU)_{x_i}$ est nécessaire car si la situation précédente était à (BTH_{x_i} - 1), l'arrivée de ce message accepté peut amener l'état des tampons à $(BTH_{x_i} - 1) + (MAXBFRU)_{x_i}$

$(MAXBFRU)_{x_i}$ est le nombre de tampons nécessaire pour prendre en charge le message le plus long, c'est-à-dire qu'il faut au moins $(MAXBFRU)_{x_i} - 1$ tampons au-dessus du BTH pour ne pas atteindre le BNO.

On pourrait croire que cela risque de retarder considérablement les demandes privilégiées.

Il n'en est rien car:

- VTAM refuse toutes les autres demandes d'entrée/sortie c'est-à-dire qu'elles seront les seules à entrer en compétition.
- son autre occupation consistera à libérer les tampons afin de rentrer dans des normes inférieures à BTH, avec pour effet de fournir des tampons aux demandes privilégiées. En fait, son taux de libération des tampons sera beaucoup plus important que celui auquel les demandes privilégiées utiliseront ces tampons.
  D'autre part, même s'il n'arrivait pas à libérer

les tampons temporairement, les délais seront minimes car, rappelons-le, les envois se limiteront exclusivement aux demandes privilégiées. Enfin, rappelons que ces tampons ne serviront que très rarement; ce qui, dans une politique plus large, rendrait des tampons, la grosse majorité du temps, inutilisés.

Nous avons introduit dans nos programmes les différents paramètres , ce qui leur permettra de calculer les $(BTH_{x_i})_{P_i}$ nécessaires aux tests.

REMARQUE: Le problème de la contrainte "nombre de tampon

L'existence de cette contrainte ne nous semble pas nécessaire si l'on applique la formule exposée précédemment. En effet, au regard de la formule, on remarque que chaque tampon supplémentaire amène une perte sèche de 72 octets, zone propre à chaque tampon.

Ceci va considérablement influencer la détermination de BSZ, car la perte d'autant de fois 72 octets va prendre une part fort significative dans la place prise.

Ce qui nous amène à dire que cette formule d'estimation d'un BSZ sur une session favorisera, dans une large mesure, la réduction du nombre de tampons, du moins jusqu'à un point réaliste.

En effet, le nombre de tampons minimum donné par un BSZ égal à la longueur maximale d'un message est inacceptable au niveau mémoire.

Autrement dit, nous pensons que cette contrainte évitera un nombre de messages susceptible d'être une source de surcharge. De plus, le rapport bénéfice tampon et bénéfice mémoire devra être largement en faveur de la place mémoire à minimiser car c'est ce qui reste, rappelons-le, notre objectif et notre

contrainte actuels.

Nous avons choisi l'objectif mémoire au détriment du côté CPU et de l'allocation des tampons.

En effet, les cas d'installation où la ressource mémoire n'est pas d'importance cruciale n'auront guère de problèmes. Il leur suffira simplement d'allouer suffisamment de places aux IOBUF.

Enfin, pratiquement, nous pourrons observer que les BSZ déterminés suivant cette méthode respecteront les contraintes employées dans les méthodes théoriques.

III.8. Intégration de plusieurs sessions

Il est bien évident que l'on ne peut se limiter à une seule session d'observation pour en tirer les conclusions générales.

On aura:

Pour chaque P_i

$C_i = \{ \bigcap (C_k)_i \}$

pour k = 1, ...,m

m est le nombre de sessions

et $(C_k)_i \in \{ U (x_i) \mid (S_{x_i})_{t_j} < ((BTH)_{x_i}) P_i$ pour tout $t_j \}$ lors de la session k

Pour chaque x_j

$(O_k)_{x_j}$

$(MAX_k)_{x_j}$

$et(O_k)_{x_j}$ est la fonction qui à un x_j fait correspondre la place prise en moyenne par seconde pour la session k.

$(MAX_k)_{x_j}$ désignera le maximum de place utilisée en octets lors de la session k pour le tampon de longueur x_j.

En résumant cette situation, chaque P_i possède son ensemble de x_i possibles.

A chaque session, et pour chaque x_j , nous possédons les données nécessaires.

III.9. Le choix des sessions

Le choix des sessions a une importance primordiale dans la validité générale de la méthode. En effet, s'il reflète la situation réelle et ses principales caractéristiques, les résultats s'adapteront au modèle futur .

C'est pourquoi, nous choisirons des situations types de l'installation. Précisons que la période des observations se fera sur des périodes de pointe, c'est-à-dire où le taux des messages est le plus élevé.

En effet, ces périodes constitueront les moments susceptibles de créer les problèmes ultérieurs.

Les sessions d'observation pourraient être choisies et obtenues par:

1. La détection des moments et des causes produisant ces pointes.
2. Reproduction de ces causes de manière suffisamment pessimiste afin de couvrir les cas les plus défavorables.

Le suivi de VTAM et les sorties de RMF, lors de ces sessions, formeront les observations qui dégageront les données nécessaires.

Nous fournirons également des renseignements qui permettront de se rendre compte à postériori de la validité de la session observée. (cfr Ch. III.11)

III.10. Détermination de la place mémoire et du BSZ

Nous possédons donc par P_i l'ensemble des x_i possibles.

Nous déterminons le P_i minimal nécessaire pour prendre en charge la situation observée:

$$P_i \text{ min} : \min_j P_j \mid C_j \neq \emptyset$$

1/ détermination du BSZ:

Pour chaque x_i, nous allons déterminer la session susceptible de poser le plus de problèmes ultérieurement, en quelque sorte, la plus dangereuse.

En d'autres termes, les autres sessions ne poseront pas de problèmes à ce BSZ.

A cette fin, nous ferons intervenir trois critères:

a/ la valeur $(MAX_x)_{x_i}$ pour chaque session

b/ la valeur $(O_k)_{x_i}$ pour chaque session

c/ la validité de la session d'observation et le profil de l'utilisation des tampons.

Ce dernier critère permettra d'augmenter ou de diminuer l'importance relative des valeurs $(MAX_k)_{x_i}$ et $(O_k)_{x_i}$.

Si, par exemple, l'on voit qu'un $(MAX_k)_{x_i}$ s'adapte mal à une pointe, l'importance de ce $(MAX_k)_{x_i}$ sera accrue par rapport aux autres. On s'aperçoit de la façon dont on arrivera à pondérer la comparaison des $(MAX_k)_{x_i}$.

Comparaison de ces paramètres:

Si , dans certains cas, on verra les différence entre $(MAX_{k1})_{x_i}$ et $(MAX_{k2})_{x_i}$ du même signe que $(O_{k1})_{x_i}$ et $(O_{k2})_{x_i}$.

Il peut ne pas en être toujours de même.

On pourrait choisir la session k telle que

$(MAX_k)_{x_j} > (MAX_{k'})_{x_j}$ pour k' = 1,...,n k' ≠ k

Supposons deux sessions k1 et k2

$(O_{k1})_{x_1}$ = 2.000 octets $(O_{k2})_{x_1}$ = 2.900 octets

$(MAX_{k1})_{x_1}$ = 3.500 octets $(MAX_{k2})_{x_1}$ = 3.499 octets

Choisir la session k1 apparait ici comme un mauvais choix car x_1 a dans la session k2 une place occupée nettement supérieure pour une différence dans les MAX insignifiante.

L'importance des différents écarts entre les $(O_k)_{x_i}$ et les $(MAX_k)_{x_i}$ devra permettre de choisir la session pour qui la combinaison des deux sera la plus exigeante.

Cependant, il pourra exister des cas où le choix sera loin d'être évident.

Nous donnerons dans ce cas la priorité à la session dont $(MAX_k)_{x_i}$ est le plus élevé. En effet, ce seront les maximums d'utilisation qui risqueront d'atteindre le BTH.

A ce stade, nous avons $\forall x_i \in C_i$, la session k désirée, et, par la même occasion $(MAX_k)_{x_i}$ et $(O_k)_{x_i}$.

Il suffit alors d'appliquer le même raisonnement que précédemment mais, cette fois-ci, dans l'autre sens afin de déterminer le x_i qui a le moins d'exigences. Ceci sera rendu possible par la comparaison entre les x_i, des différents $(MAX_k)_{x_i}$ et $(O_k)_{x_i}$ obtenus dans les sorties de la session k déterminée précédemment pour cet x_i, en faisant également intervenir le troisième critère.

2. Détermination du BNO:

Le nombre de pages allouées sera calculé à partir du P_i choisi précédemment. On pourra éventuellement procéder à une augmentation de la taille mémoire allouée.

Elle traduirait simplement la marge de sécurité vis-à-vis de situations dont on ne saurait prévoir toutes les variations.

Cette augmentation dépendrait:

- de la validité de la session
- du comportement des tampons à l'égard de la session.

Lors de certains intervalles, la validité de la session pourra être estimée à partir de:

- la liste des applications et leurs caractéristiques
- la distribution de la longueur des messages

Le comportement des tampons sera connu à partir du profil d'utilisation défini précédemment (cfr III.6.5.).

Un exemple de raisonnement possible serait le suivant: lors d'une pointe, on s'est rendu compte qu'elle pourrait être plus accentuée; dès lors, la bonne ou mauvaise façon de réagir des tampons permettra d'estimer la mémoire supplémentaire à allouer.

Ce genre d'estimation globale ne sera possible que pour une personne connaissant les problèmes et le comportement de l'installation.

III.11. Renseignements supplémentaires:

Afin de faciliter la tâche des personnes devant procéder à certains choix, nous avons tenu à fournir des renseignements supplémentaires:

Ces renseignements seront:

- par application :
  - taux des messages
  - longueur moyenne
  - nombre
- sur toute la session: idem

III.12. Le nombre de tampons chez PPBUF.

Le BSZ est donc fixé par celui des IOBUF.

Avant de passer à la définition du BNO, examinons auparavant la validité du BSZ pour les PPBUF :

- ils ne contiennent que des messages d'entrée qui n'ont pas à priori la même distribution que ceux de sortie.
- dans le cas où des envois sont faits de la part d'un terminal (batch ou autre) et que l'application n'est pas en état de les traiter, la distribution des messages s'écartera de l'initiale vu que les messages vont s'accumuler.

Mais d'autre part,

- c'est au niveau des IOBUF que se passera le ralentissement, c'est-à-dire que leur situation à ce moment-là influencera le résultat.
- en raison des différents sens que peut prendre ce changement de distribution, un BSZ adéquat dans un cas ne le serait pas dans l'autre et nous retomberions devant le même problème.
- favoriser quand même le PPBUF nous amènerait à créer des problèmes dans les IOBUF.

Nous ne trouvons aucun avantage à reconsidérer la valeur BSZ préalablement définie.

$$BNO = \left(\sum_{i=1}^{n} A_i \cdot B_i \right) / BSZ$$

où n représente le nombre de terminaux en entrée ,

A_i représente la longueur maximale du message que le terminal i peut envoyer,

B_i est le nombre maximum de messages que ce terminal peut envoyer.

Rappelons que ces deux données sont définies à VTAM.

B_i sera égal à 1 pour les terminaux non batch.

La taille résultant de cette formule permettra de supporter les envois maximum de tous les terminaux batch. Elle permettra également de prendre en charge le cas où tous les autres terminaux enverraient leur message de longueur maximale.

Nous considérons que la taille résultant de cette formule sera suffisamment élevée car:

- les terminaux enverront rarement leur message de longueur maximale
- le nombre de terminaux actifs sera rarement maximum

mais d'autre part:

- les envois d'un terminal batch occuperont souvent le maximum de tampons qu'il peut utiliser dans les PPBUF
- un terminal pourra envoyer plus d'un message
- le coût de la mémoire paginable est nettement moins élevé
- un ralentissement dû au PPBUF sera, la plupart du temps, irrémédiable.

REMARQUES GENERALES

A. si la situation réelle est en ralentissement:

Un problème peut se poser si la méthode est appliquée à postériori. En effet, si le BTH est atteint, l'arrivée des messages sera fortement influencée vu que VTAM n'acceptera plus que les demandes privilégiées.

Une autre longueur de tampon déterminée par notre méthode se serait peut-être limitée à retarder l'échéance. La solution serait d'augmenter le BNO existant

de manière temporaire afin d'éviter ce genre d'ennui.

B. Charge du"suivi"de VTAM.

L'exécution par VTAM du "suivi" aura deux conséquences:

- ralentissement de la fréquence d'arrivée des messages
- ralentissement également dans la libération des tampons,

c'est-à-dire que ces deux conséquences auront tendance à s'éliminer au niveau des contraintes.

Le ralentissement, quant à lui, est dû au fait que VTAM dispose de moins de temps:

- pour répondre aux terminaux
- libérer les IOBUF.

oooooooooooooo

CAS PRATIQUE

Nous avons eu à notre disposition une bande du suivi VTAM.

Elle était le résultat de la simulation d'une configuration de 36 terminaux TSO.

Elle contenait 5200 messages pour un intervalle de 30 minutes, c'est-à-dire un taux d'arrivée extrêmement élevé.

Malheureusement, nous ne pouvions avoir à notre disposition l'état exact des tampons réellement utilisés.

Ceci nous a amené à nous donner des valeurs réalistes et à déterminer ainsi les P_{x_i} qui, malheureusement, ne reflètaient pas exactement la réalité.

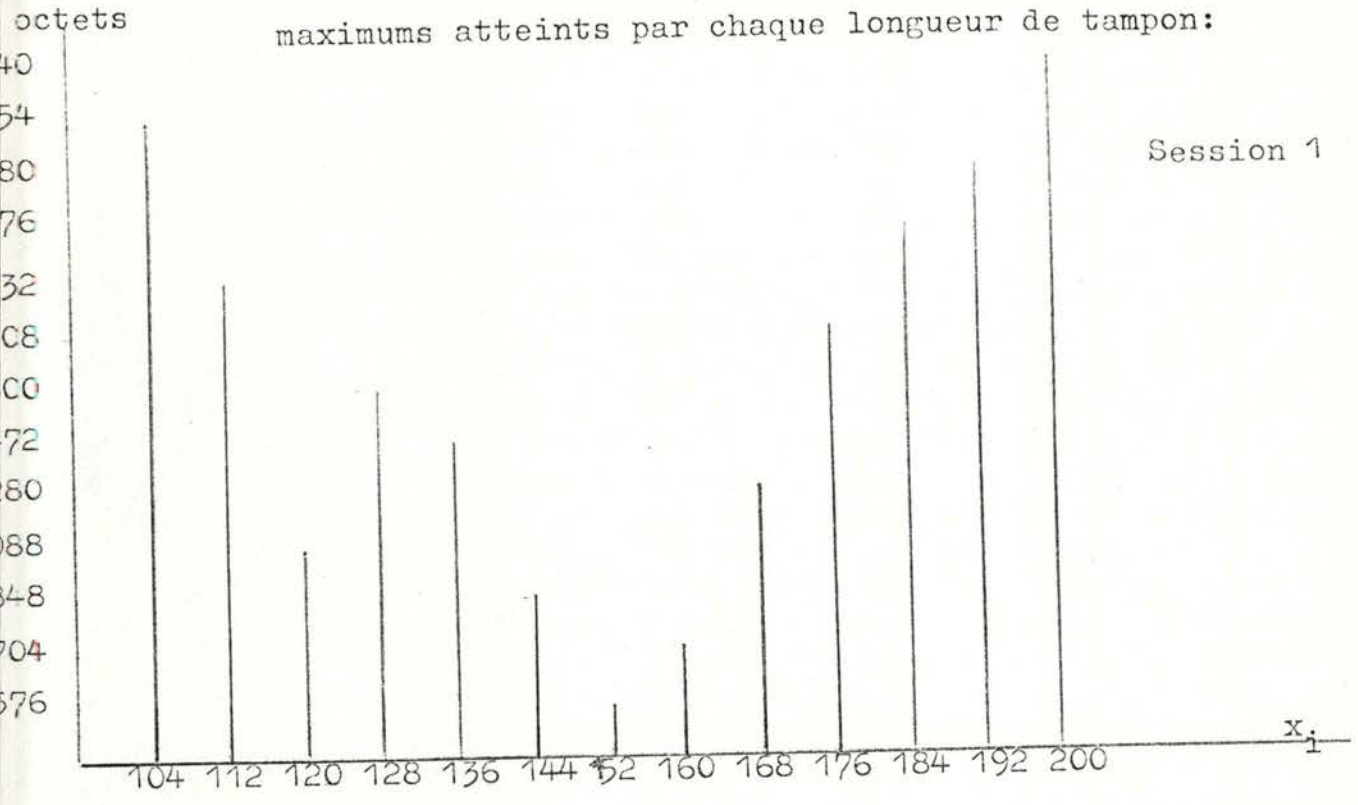
Cependant, nous ne pensons pas que cette exigence diminue dans une quelconque mesure l'intérêt de la méthode. En effet, ces chiffres seraient le résultat d'une observation aisée grâce à nos programmes tournant dans RMF.

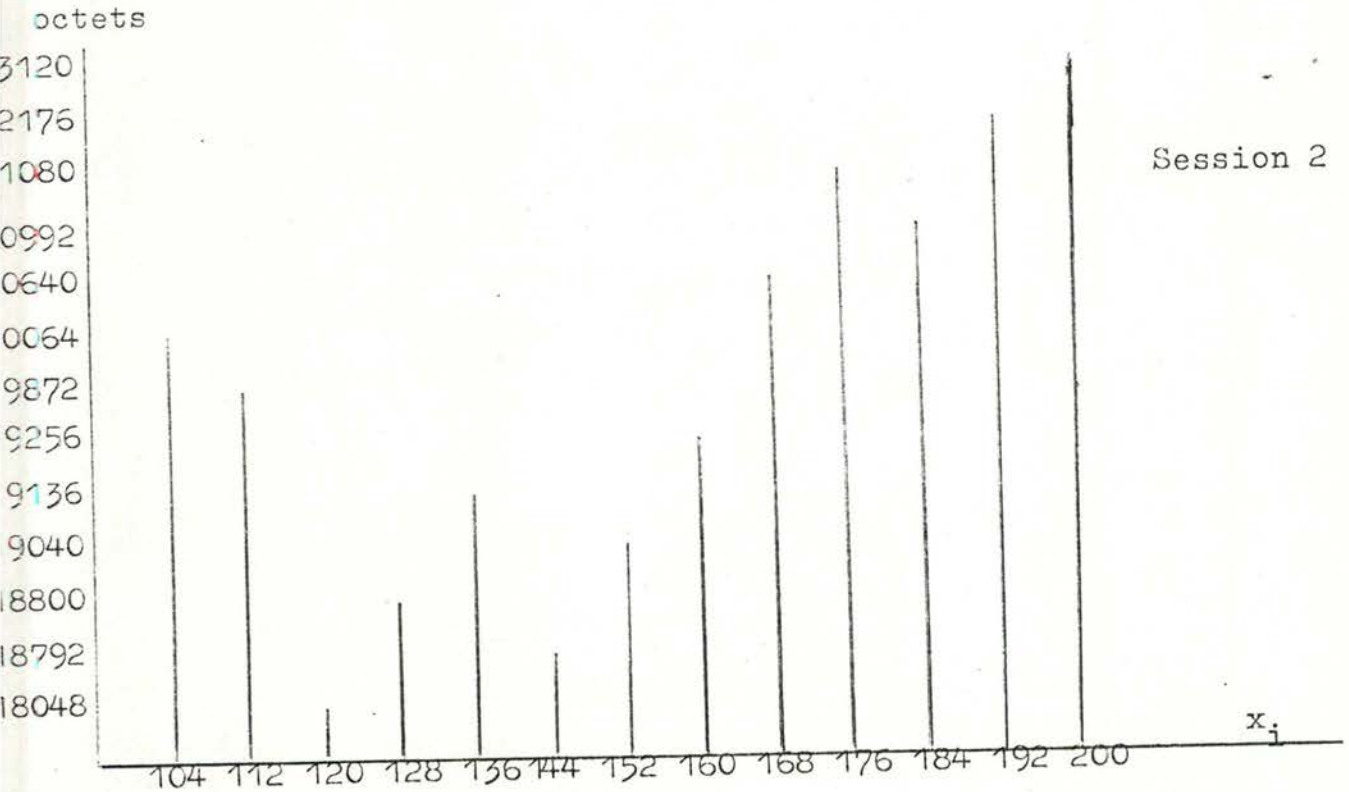
Connaissant cependant le BSZ de l'installation, il nous sera possible de faire une comparaison entre le BSZ déterminé par notre méthode et le BSZ réel.

Ceci nous permettra d'avoir une idée du gain mémoire acquis grâce à un nouveau BSZ.

En pratique, nous avons supposé que la bande de simulation était composée de deux sessions d'observation. Les P_{x_i} étaient donc différents dans les deux.

Si l'on représente par graphique les différents maximums atteints par chaque longueur de tampon:





$(C_1)_{P_5}$ = {112, 120, 128, 144, 152, 160, 168, 176, 184, 192, 200}

Les $(O_1)_{x_i}$ sont croissants avec la valeur de x_i , c'est-à dire que plus le x_i est grand, plus $(O_1)_{x_i}$ est élevé. Les différences sont de l'ordre de 20 à 25 octets entre chaque x_i (cfr listing)

$(C_2)_{P_5}$ = {120, 128, 144, 152}

P_i min = P_5 c'est-à-dire que 5 pages auraient suffi.

$(C)_{P_5}$ = {120, 128, 144, 152}

Précisons également que les $(O_2)_{x_i}$ sont également croissants avec la valeur de x_i avec des différences de 10 à 15 octets.

Pour chaque x_i la session deux est la plus dangereuse.

120 est le tampon qui a le moins d'exigence; en effet, son MAX est le plus petit et son $(O)_{120}$ l'est également dans les x_i possibles.

Analysons la distribution de la longueur des messages :

80% des messages ont une longueur $\leqslant$ 80 octets

4% des messages ont une longueur $\geqslant$ 301 octets

Le reste, bien sûr se situe entre les deux .

A cette condition, comment peut-on concevoir qu'un tampon de 120 octets de long puisse avoir donné le min (max_{x_i}) absolu, alors qu'il a, par rapport aux tampons plus petits, une moyenne d'utilisation par seconde de l'ordre de 10 à 15% inférieure.

A cet effet, supposons que:

U (80) : 40

U (120) : 34

U (x_i) le nombre de tampons actuellement utilisés

Soit un message qui arrive de longueur de 1400 octets.

U (80) : 40 +(f(1400/80)) = 58

U (120) :34 + (f(1400/120) = 45

On voit donc l'importance des messages de grande longueur sur la situation réelle et comment la situation décrite plus haut peut se produire.

Le gain vis-à-vis du tampon existant (112) est de 2K.

Ces deux K, cependant, amèneront une différence de 1 page, lors de la réservation finale.

En effet, si l'on réserve 6 pages, cela risque d'être insuffisant car les six pages - $(BNO-BTH)_{112}$ nous amèneront à une réserve insuffisante en fonction du maximum. C'est pourquoi, si ce tampon avait été choisi, 7 pages auraient été nécessaires.

Les définitions résultant de la simulation seront:

BSZ : 120 BNO: 128 BTH: 115

oooooooooo

CHAPITRE IV : S.R.M.

IV.1 INTRODUCTION

Le SRM est une nouvelle caractéristique d'un système d'exploitation en ce sens qu'il a remis au sein d'une même composante des algorithmes de performance jusqu'ici séparés.

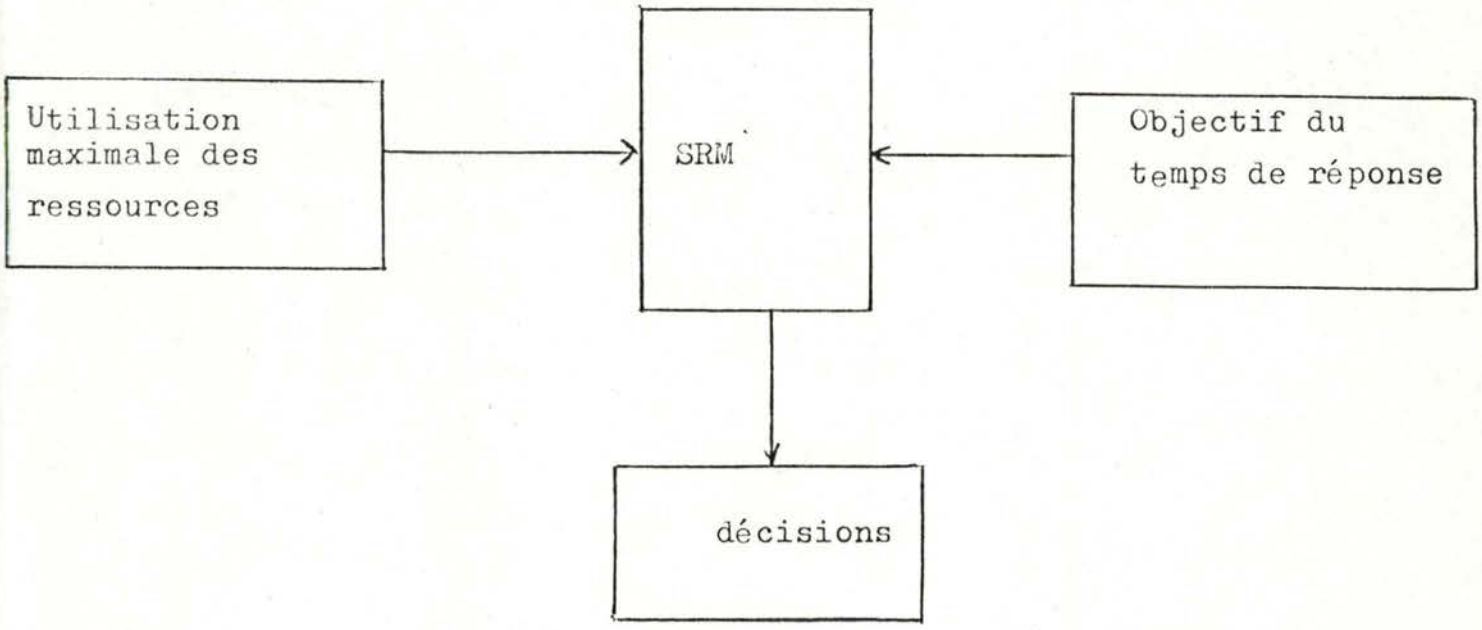
Certains nouveaux côtés ont été introduits notamment afin de permettre un contrôle plus développé sur la charge du système.

Les objectifs de SRM (system ressources manager) sont globalement les suivants:

-utilisation optimale des ressources

- distribution des ressources de manière à respecter les exigences de temps de réponse

SCHEMA I:



Pour arriver à ses fins, SRM procédera à des décisions rencontrant ses objectifs.

Les décisions s'effectuent de la manière suivante:

- écriture de la zone de travail d'un processus sur mémoire auxiliaire (que nous appelerons désormais expulsion)
- calcul de la priorité dynamique des processus
- réintroduction d'un processus expulsé

Le principe de départ est de partager tous les genres de processus en domaines. Chaque domaine regroupera les processus possédant les mêmes caractéristiques d'exécution.

En effectuant un contrôle sur les niveaux de multiprogrammation à l'intérieur de chaque domaine grâce à ces décisions, SRM tentera de garder un ensemble de travaux adéquat.

Par exemple, une façon de découper sera de regrouper séparément les courts TSO, les batchs prioritaires, les batchs non prioritaires.

SRM choisira le meilleur processus pour être réintroduit ou être expulsé, de manière à:

- optimaliser l'utilisation des ressources
- satisfaire les objectifs de temps de réponse de l'utilisateur

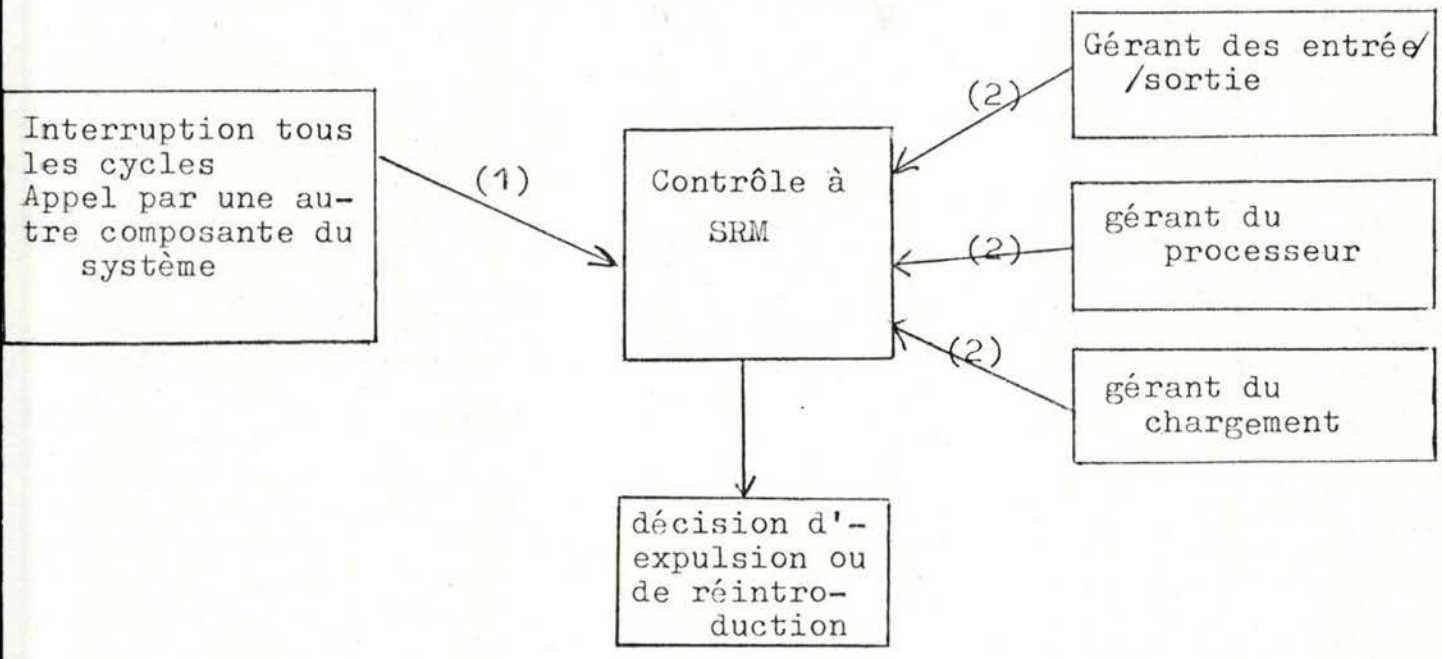
Cette décision d'expulsion ou de réintroduction se fera sur base de l'utilisation globale des ressources.

Les ressources générales sont :

- le processeur
- le matériel pour les entrées / sorties
- la mémoire centrale

La décision d'expulsion de SRM reviendra en pratique à empêcher le processus d'utiliser les ressources.

IV.2 . Diagramme général de fonctionnement



(1) signifie que SRM prendra la main

(2) les différents algorithmes correspondant aux différents gérants donneront chacun des valeurs de recommandation pour un processus. Ces valeurs représenteront les conseils de chaque gérant.

SRM se basera sur les valeurs données par les gérants pour ses différentes décisions d'expulsion ou de réintroduction .

Le system ressource manager choisira un niveau de multiprogrammation (MPL) en se basant sur:

1. la situation d'ensemble d'utilisation des ressources.
2. Les contraintes spécifiées dans un domaine particulier.
3. Les situations particulières des processus à l'intérieur d'un domaine en relation avec leurs objectifs.

Les contraintes spécifiées dans un domaine sont les suivantes:

MIN MPL : le nombre minimum de processus dans ce domaine.

MAX MPL: le nombre maximum.

le poids du domaine: équivalent en quelque sorte à son importance.

Le point 1 sera spécifié grâce aux gérants des ressources qui donneront à SRM des valeurs générales d'utilisation des trois ressources.

Les gérants de la charge, du processus, des entrées/sorties donneront des valeurs de recommandation valables à l'intérieur d'un domaine et concernant chaque processus appartenant actuellement à ce domaine.

Le processus général d'une décision de SRM est le suivant:

- choix d'une expulsion ou d'une réintroduction suivant l'utilisation des trois grandes ressources
- choix d'un domaine
- calcul des valeurs de recommandation pour chaque algorithme pour tous les processus appartenant au domaine
- cumul des différentes valeurs et choix du ou des processus sur lequel se portera la décision

Il faut donc remarquer que le calcul des valeurs de recommandation par les gérants du processeur et des entrées/sorties ne se fera que pour les processus appartenant au domaine choisi.

IV.3. Le gérant du processeur

Il se subdivise en trois grandes fonctions:

1° une fonction d'équilibre de l'utilisation du processeur.

2° calcul de la priorité dynamique

3° Contrôle des processus utilisant une ressource protégée.

IV.3.1. fonction d'équilibre

Cette fonction donnera les valeurs de recommandation. Elle permettra de contrôler:

- l'utilisation générale du processeur
- l'utilisation individuelle du processeur par processus.

Elle possède donc tous les renseignements nécessaires (qu'elle calcule) afin d'estimer toutes ces utilisations.

La valeur de recommandation pour un processus particulier se calculera comme suit:

1) Si le processeur est sous-utilisé(temps d'attente supérieur à une constante), alors les gros utilisateurs vont recevoir une valeur de recommandation positive proportionnelle à l'importance de cette sous-utilisation et à la proportion dans laquelle le processus peut y remédier, c'est-à-dire que plus il sera un important utilisateur du processeur, plus il recevra une recommandation importante.

A ce stade, il faut déjà faire une remarque valable pour les autres algorithmes: Les valeurs de recommandation à l'intérieur d'un domaine sont calculées pour tous les processus dans le domaine, qu'ils soient expulsés ou qu'ils soient actifs, cela implique que SRM garde des informations sur la façon d'utiliser les ressources de la part du processus.

En pratique la recommandation se fera pour un

processus détecté comme un gros utilisateur de processeur, par la comparaison du taux auquel il utilise le processeur par rapport à une constante définie. Les processus non détectés recevront une recommandation nulle.

2) Le processeur est sur-utilisé:

Dans ce cas, les utilisateurs détectés comme lourds recevront une recommandation négative tandis que celle des autres sera nulle.

De manière générale, la recommandation est, pour un processus:

> 0 (réintroduction) si sous-utilisation

< 0 (expulsion) s'il y a sur-utilisation

= 0 s'il n'a pas été identifié.

IV.3.2. Calcul de la priorité dynamique

En pratique, SRM exerce le contrôle sur la priorité des processus, dont on a spécifié une priorité qui tombe dans celles contrôlées par SRM (il ne les contrôle donc pas toutes).

SRM exerce son contrôle sur un ou plusieurs groupes de 16 priorités dans lesquelles il va calculer la priorité dynamique, suivant trois algorihmes:

- temps moyen d'attente (MTW)
- les priorités fixées (F)
- les priorités rotatives (R)

Ces trois algorithmes s'appliqueront à l'intérieur d'un groupe de 16 priorités.

Ce groupe de 16 priorités est divisé en trois ensembles correspondant aux trois algorithmes et chaque algorithme s'appliquera à un ensemble et calculera la priorité qui devra toujours appartenir au même ensemble.

Par exemple, si la priorité d'un utilisateur est 35

et que le groupe 32 à 48 est contrôlé par SRM, que, de plus, le sous-ensemble 32 à 38 est le groupe temps moyen d'attente, la priorité dynamique de ce processus variera toujours de 32 à 38. Le découpage en trois ensembles est commun à tous les groupes (ex. MTW : les 8 premières)

- l'algorithme MTW favorisera les processus à orientation entrée/sortie.
- le groupe de priorités rotatives est un groupe où tous les x temps, SRM mettra le dernier en tête du groupe et le premier en queue.
- les priorités fixées ne changent pas.

IV.3.3. Protection des processus utilisant une ressource privilégiée.

On spécifie à SRM un intervalle de temps pendant lequel un processus utilisant une ressource en la réservant ne peut être expulsé, ceci afin d'éviter que les autres processus ne soient bloqués jusqu'à ce que le premier libère la ressource. En effet, s'il est expulsé de la mémoire, les processus désirant cette ressource ne pourront recevoir l'autorisation.

IV.4. Gérant des entrée/sortie.

Il se compose d'une fonction dont le but est d'équilibrer l'utilisation au niveau des canaux logiques.

Les recommandations se feront donc de la même manière que pour le gérant du processeur, mais cette fois-ci sur base de chaque canal logique.

Un canal logique sera considéré comme sous-utilisé ou sur-utilisé sur base de limite d'un taux d'EXCP par seconde.

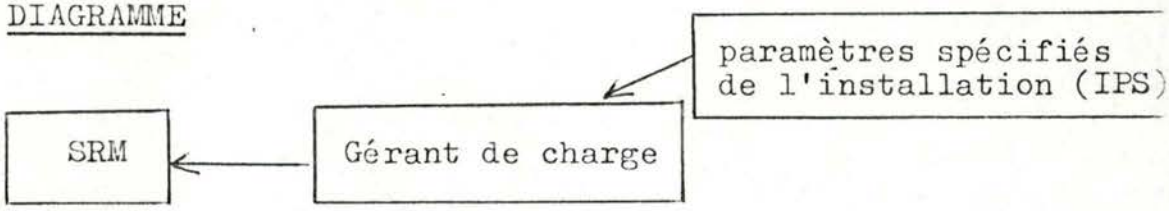
Les processus seront identifiés de la même manière.

IV.5. Le gérant de la charge.

Rappelons que toutes les valeurs de recommandation se feront à l'intérieur d'un domaine.

Cette partie de SRM supportera les objectifs de performance de l'installation pour les sous-ensembles individuels d'utilisateurs (les domaines).

DIAGRAMME



chronologiquement, le gérant effectue les opérations suivantes:

1/ il cumule les unités de service pour tous les processus.

2/ il détecte les changements de période dans la vie d'un processus.

3/ il calcule les taux de services pour tous les processus appartenant au domaine choisi.

4/ Une fois le domaine choisi, il compare la situation des différents processus.

5/ il donne une valeur de recommandation pour chaque processus.

IV.5.1. Les unités de service.

Les unités de service sont une combinaison linéaire des trois ressources de base .

Les unités de service reçues pour un processus sont égales à:

$$A\ (\text{temps de processeur})_K + B\ (\text{nombre d'EXCP}) + C(\ \text{nombre moyen de page pour sa zone de travail} * \text{temps processeur})$$

$$\frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{K}$$

où K est un facteur d'échelle
et A, B, C sont des coëfficients qui représenteront l'importance accordée à une ressource particulière.

IV.5.2. Détection des périodes :

A la vie d'un processus, correspondent plusieurs objectifs de performance. Nous verrons plus tard où interviendra ce problème.

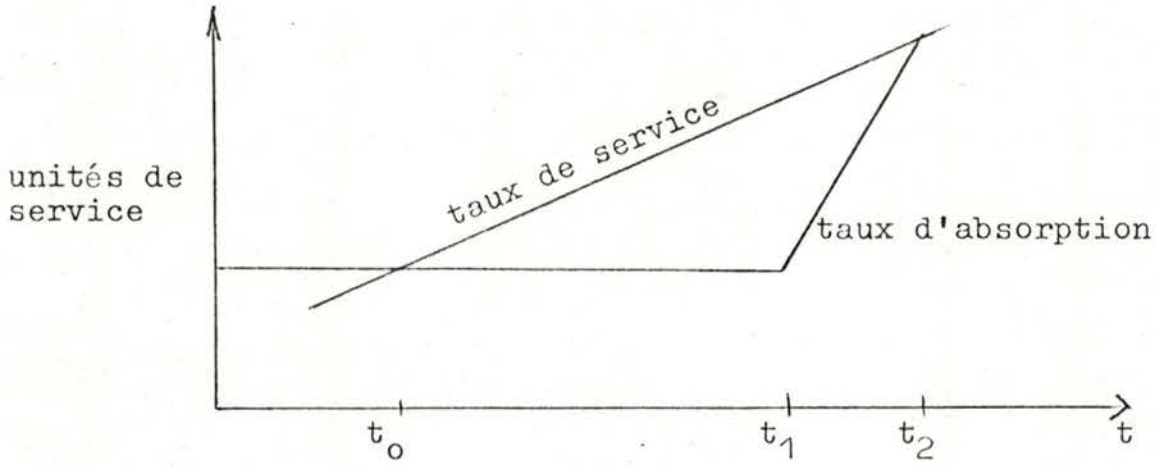
IV5.3. Calcul des taux

Le gérant va calculer deux taux:

- le taux d'absorption: c'est le taux auquel le processus reçoit les unités de services quand il est en mémoire principale. Il est égal au total des unités de services reçues quand il est actif, divisé par le temps où il est actif ; il est donc à zéro quand le processus est expulsé.

- le taux de service: c'est le taux auquel un processus reçoit les unités de service depuis le démarrage; il est donc égal au total des unités de service reçues sur le temps écoulé depuis son lancement.

Si l'on représente ces deux taux par une fonction:



t_o représente le moment où le processus est expulsé
t_1 représente le moment où il est réintroduit
le moment t_1 à t_2 et les unités des services cumulés vont donner le taux d'absorption tandis que le moment (t_o , t_2) va définir le taux de service sur les deux périodes.

Sur l'intervalle (t_1,t_2) , la pente du taux de service est moins importante que celle du taux d'absorption vu qu'il ne reçoit pas d'unité de services pendant l'intervalle (t_o, t_1) . Le taux de service représentera donc le taux général auquel il a eu accès aux ressources depuis le début.

IV.5.4 Comparaison.

Lorsqu'on procède à une décision à l'intérieur d'un domaine, il est nécessaire de comparer les différents processus afin de choisir l'adéquat.

Une nouvelle notion, la charge correspondant à un processus, rendra cette comparaison possible.

Le concept est de donner une unité de commune mesure à tous les processus dans le domaine; il représentera la

façon dont il exploite les ressources pour son pouvoir à les utiliser.

La formulation sous forme de charge sera une manière de répondre à la question: "Donnez-moi une quantification de la façon dont le processus utilise les ressources".

La fonction qui associera à un taux de service sa charge correspondante sera précisée grâce à un objectif de performance qui associe des charges à des valeurs de taux de service.

Nous pensons qu'il sera plus aisé de saisir le principe de raisonnement sur un exemple.

Supposons deux objectifs de performance:

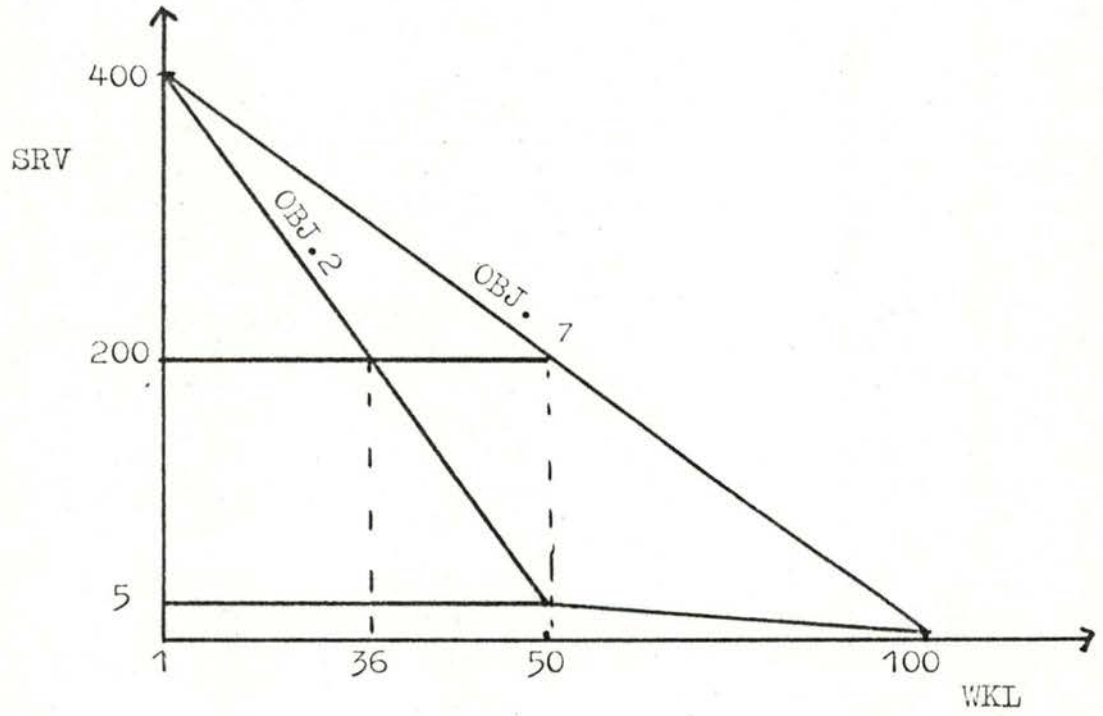
objectif 1 est défini par SRV:(400,200,0) et WKL(1,50,100)
et obj1 va être déterminé par les trois couples
(400,1) (200,50) (0,100)

objectif 2 est défini par SRV:(400,5,0) et WKL(1,50,100)

SRV représente des taux de service

WKL des valeurs de charge

Voici le graphique des deux fonctions



Supposons maintenant deux processus ayant chacun Objectif/1 et objectif/2 respectivement comme objectif de performance.

Supposons également qu'ils aient tous les deux un taux de service de 200.

Pour le processus 1 WKL = 50

Pour le processus 2 WKL = ± 36

c'est-à-dire que le processus 1 a plus droit d'avoir un taux de service de 200 que le processus 2.

IV.5.5. Calcul des recommandations.

Les valeurs de recommandation seront les WKL associés au SRV ou dans le cas précédent 50 et 36.

En d'autres termes, les valeurs fournies par le gérant de la charge seront une estimation du droit du processus à avoir le taux de service qu'il emploie.

Une remarque importante doit être faite:

C'est que les processus nouveaux, vu qu'ils n'ont pas encore reçu de services, ont un taux de service égal à zéro, ce qui donnera un WKL maximum.

A l'intérieur d'un domaine, il s'en suivra généralement: l'expulsion d'un processus qui vient de commencer, et l'introduction du nouveau. Ceci sera évité grâce à un nombre d'unités de services que l'on précise à SRM et avec comme effet pratique d'empêcher ce processus d'être expulsé tant qu'il n'aura pas reçu un nombre d'unités de service égal au nombre précisé.

Il reste maintenant à inclure le point"période" dans tout ceci ou, du moins, l'expliquer, car, chronologiquement, il était à sa place.

Le principe est le suivant: on associe à différents moments de la vie d'un processus, des objectifs de performance différents et également des domaines différents.

En effet, les caractéristiques d'exécution d'un processus peuvent changer de telle manière que:

- ses exigences ne sont plus les mêmes
- son appartenance au même domaine ne se justifie plus

La vie d'un processus sera divisé en plusieurs périodes et à chaque période sera associé un objectif de performance adéquat, ainsi qu'un domaine respectant ses caractéristiques.

L'ensemble des périodes, et associé à chaque période son objectif de performance, sa durée, et son domaine sera précisé grâce à la définition d'un groupe de performance qui aura la configuration suivante:

période 1: durée, OBJ N_1 , DOMAINE M_1

période 2: durée, OBJ N_2 , DOMAINE M_2

La précision d'un groupe de performance lors de l'introduction d'un programme dans le système permettra de déterminer tous les éléments dont SRM se servira pour régler son exécution.

Enfin, le gérant détecte la fin d'une période en regardant si le cumul des unités de service ne dépasse pas celui indiqué pour la période ; la durée d'une période est précisée en unités de service. Si cette limite est atteinte, il passera à la période suivante en tenant compte des caractéristiques différentes, s'adaptant à la nouvelle.

IV.6. Situation des paramètres:

Tous les paramètres se situent dans l'IPS, du moins ceux cités précédemment, sauf, pour chaque ressource, les coëfficients intervenant lors du calcul des unités de service qui se situent eux sur une autre librairie accessible à SRM.

Il reste à mentionner l'existence d'un dernier paramètre qui prendra deux valeurs : 0 ou 1 (RTB : response/throughput/bias).

0 signifie que SRM ne tiendra pas compte des recommandations des gérants du processeur et des entrée/sortie et 1 qu'il en tiendra compte.

Le RTB à 0 favorisera l'objectif temps de réponse, au détriment de l'objectif utilisation des ressources.

IV.7. Les façons dont SRM prend la main:

Nous avons exposé précédemment que SRM pouvait obtenir la main de deux façons:

- par des interruptions tous les x temps.
- par un appel d'une autre partie du système.

IV.7.1. Par interruption .

SRM prendra le contrôle toutes les x secondes et procédera au choix d'un domaine où il exercera ses décisions (variables en fonction de la charge).

IV.7.2. Appel par une autre partie (SYSEVENT)

En effet, une autre partie du système peut demander certains services à SRM. Un exemple pratique est le cas produit par une pénurie de pages en mémoire réelle. Le gérant de la mémoire avertira SRM de l'état actuel afin qu'il prenne des mesures.

Outre l'expulsion de mémoire réelle de certains processus, SRM réagira en empêchant la création de nouveaux espaces d'adresses.

Ceci est rendu possible, entre autre, par le fait qu'une commande "start" à la console provoquera l'appel de SRM qui, en retour, autorisera ou non la demande. Dans le cas où le nombre de pages réelles disponibles est tombé en-dessous de la limite, SRM répondra non.

IV.8 . Choix d'un domaine:

SRM choisira le domaine possédant la valeur suivante la plus élevée.

$$C = \frac{\text{nombre moyen(1) de processus prêts x poids du domaine(2)}}{\text{nombre actuellement actif (3)}}$$

Si on étudie (1) sur (3), on remarque que s'il y a plus de processus prêts qu'il n'y en a en moyenne, le rapport (1) sur (3) sera inférieur à 1 et diminuera ainsi l'importance du domaine. La tendance générale favorisera les domaines prioritaires, du moins jusqu'à ce que leur nombre actif devienne prépondérant.

D'un autre côté, le rapport (1) sur (3) supérieur à 1 augmentera l'importance du domaine.

IV.9. Décision d'expulsion ou de réintroduction.

Les décisions d'expulsion ou de réintroduction se feront sur base de valeur d'utilisation idéale des trois grandes ressources qu'on lui donne et sur l'utilisation actuelle des ressources.

Il décidera d'une expulsion lors de sur-utilisation de certaines ressources et d'une réintroduction en cas de sous-utilisation.

Eventuellement, il choisira de faire les deux, puis, une fois son choix fixé, il décidera du domaine où exercer ses décisions.

IV.10. La Charge de SRM.

Nous allons distinguer deux sortes de charges:
- celle due à l'exécution des programmes de SRM eux-mêmes.
- celle due à l'expulsion du processus.

IV.10.1. SRM lui-même

C'est une charge très difficile à estimer et nous ne disposons pas des outils nécessaires à son évaluation. En pratique, la tendance générale est que SRM prend la main moins souvent par des interruptions au fur et à mesure que la charge générale augmente. (la charge de SRM est évaluée à 5% du temps du processeur, sur une charge normale).

IV.10.2. L'expulsion du processus ou la réintroduction.

On peut subdiviser cette charge en trois ensembles.

1/ le temps nécessaire à une tâche spéciale qui appartiendra au processus pour se préparer à l'expulsion.

2/ le temps nécessaire pour traiter les interruptions d'entrée/sortie afin de transférer les pages de la zone de travail vers les supports nécessaires.

3/ le temps nécessaire au gérant des mémoires pour mettre ses tables à jour.

Enfin, SRM nécessite l'existence d'une unité à accès direct dont l'usage sera de garder les pages des processus expulsés.

IV.11. En pratique.

Tout au long de cet exposé, nous nous sommes aperçu de la complexité de la définition des paramètres à SRM.

Cette complexité imposera de longs tâtonnements afin d'obtenir des définitions susceptibles de satisfaire les objectifs de l'installation.

Cette définition adéquate ne sera d'ailleurs rendue possible que par l'intermédiaire d'un outil de mesure : RMF

Cet outil analysera l'utilisation des ressources du système du point de vue essentiellement orienté SRM, par exemple: il exprimera le nombre d'unités de services rendus à un processus, ce qui permettra de vérifier la bonne application et l'effet pratique de ceux définis à SRM.

En pratique, RMF ne fera qu'observer des zones données qu'SRM crée et met à jour.

On pourrait croire, en pratique, qu'une bonne partie des raisons de réintroduction et d'expulsion sera due au gérant des charges.

Les causes de ces décisions et leur taux font partie d'un rapport fourni par RMF.

Or, il nous a été donné de pouvoir observer certains de ces rapports concernant de très grosses entreprises.

Dans ces rapports, on pouvait constater qu'une énorme majorité (95%) des causes d'expulsion et de réintroduction étaient dues à un temps d'attente d'évènement d'entrée/sortie (ex. attente de la réponse d'un terminal).

Ce qui revient à dire que l'utilisation des capacités de SRM est fortement réduite et ne comporte pratiquement qu'une caractéristique existant déjà sur les systèmes précédents.

Cette décision d'expulsion est d'ailleurs provoquée par un appel à SRM lorsque le système détecte un temps d'attente suffisamment significatif.

Le problème est dû, à notre avis, à la complexité engendrée par la détermination des paramètres et à l'estimation de leur interaction.

En effet, s'il semble relativement aisé de définir les objectifs pour un processus, évaluer les conséquences des interactions nous semble nettement plus aléatoire .

Si les paramètres sont extrêmement difficiles à cerner, il est évident que, bien utilisé, SRM fournira l'outil adéquat qui permettra de contrôler et de respecter les objectifs de l'installation d'une manière extrêmement précise.

oooooooooooooo

CHAPITRE V : RMF et nos outils de mesure.

V.1. INTRODUCTION.

Nous avons donc écrit les programmes qui fonctionnent dans le cadre de RMF et qui ont pour but de sortir des renseignements sur l'état des tampons de VTAM.

Or, RMF est un outil essentiellement destiné à supporter SRM.

C'est pourquoi, outre le fait que SRM est une nouvelle caractéristique, nous avons tenu à définir les grandes lignes de ce composant important du système.

A présent, nous allons exposer brièvement RMF pour ensuite passer à une description plus détaillée de nos programmes.

V.2 - RMF (ressource measurement facility)

V.2.1. Fonction et méthode d'approche.

Le "ressource measurement facility" de MVS est un moyen de mesurer les performance du système et de montrer les différentes sources de tous les problèmes afin de contrôler les paramètres qui régissent SRM.

Un contrôle effectif du système demande des mesures sur de longues périodes qui soient faciles à utiliser et capables de donner les renseignements désirés à différents niveaux de détails.

Les données obtenues devraient aider, en relation avec SRM, au contrôle de la performance et des objectifs demandés.

Une méthode d'approche pourrait être:

1/ Identifier les composantes du système qui ont un pourcentage d'utilisation exceptionnel. Les critères de qualification employés par RMF sont standards et correspondent lorsque, c'est le cas, aux mêmes genres de données que pour la définition des paramètres dans SRM (ex. unités de service).

2/ Identifier les moments où l'utilisation de telle ou telle ressource est exceptionnelle.

3/ Faire la correspondance entre les services alloués aux utilisateurs et les services théoriques dans les circonstances d'un niveau de charge particulier spécifié dans l'IPS; autrement dit, si l'espace d'adresse reçoit bien ce qu'il devrait recevoir.

4/ Identifier les embouteillages.

5/ Identifier les utilisateurs excessifs de certaines ressources.

On détectera les problèmes au niveau le plus général, puis on lancera les mesures dans des domaines bien particuliers.

Le genre de mesures générales regroupe les suivantes: processeur, pagination, charge, canaux, l'activité des unités à accès direct.

Les mesures de la charge, afin de permettre la vérification des paramètres précisés à SRM, sont disponibles soit par domaine, groupe de performance, objectif de performance.

Ces mesures fournissent une bonne représentation de l'activité du système pour des périodes relativement longues.

On peut donc alors initier les rapports à des niveaux plus détaillés.

V.2.2. Méthodes d'utilisation:

Il existe deux moyens d'utiliser RMF.

A ces deux moyens, correspondent les deux types de mesure possibles : générale (session de type 1) ou détaillée (session de type 2).

La session de type 1 sera démarrée à la console et supportera, suivant les options précisées, des mesures globales sur le système. Ces mesures globales sont le résultat de la réduction des données collectées au cours d'un intervalle d'observation (MIN, MAX,MOYENNE).

A l'intérieur d'un intervalle, l'option cycle définira la périodicité d'observation des données qui seront réduites par après.

La session de type 2 fournira des données détaillées et non réduites au terminal. Elle permettra de suivre "en direct" l'évolution des données désirées.

C'est en quelque sorte un session instantanée.

V.2.3. Conclusion sur RMF:

Cet outil de mesure, qui ne fait qu'observer des zones de données appartenant à SRM, offre donc de larges possibilités pour la mesure de la performance du système.

En relation avec SRM, il pourra aider, dans une grande part, à améliorer cette performance.

V.3. Nos outils.

Comme nous l'avons dit, RMF est surtout l'outil sans lequel SRM ne serait qu'une utopie.

De plus, on considérait comme ressources du système:le processeur, les entrées/sorties, ainsi que les mémoires.

La gestion des tampons appartenant à cette dernière ressource peut revêtir une importance particulière.

Ces tampons réservés à VTAM ne faisaient pas l'objet d'une observation par RMF. Nous avons écrit une série de routines dont le but est d'observer des renseignements qui donnent l'état des tampons VTAM.

Ces routines ont été incluses à RMF et sont devenues une option supplémentaire pour l'utilisateur qui le désire.

Comme les deux types de mesures possibles (générale ou détaillée) ont chacun leurs avantages,nous avons inclus deux séries de programmes: la première s'exécutera sous le contrôle de la session de type 1 et la second sous le contrôle de la session de type 2.

V.3.1. Sous le contrôle de la session de type 1.

Les programmes qui s'exécutent sous le contrôle de la session 1 sont au nombre de cinq et à chacun correspondent des fonctions bien particulières, comme dans le cadre de RMF dont nous allons voir brièvement l'organisation générale pour la session 1.

RMF (1) occupe donc un espace d'adresse et se compose de ces grandes sortes de programmes:

1. les programmes de contrôle.
2. les programmes d'initialisation.
3. les programmes d'observation.
4. les programmes de traitement de fin d'intervalle.
5. les programmes de rapport.
6. les programmes de clôture.

La séquence des opérations est le suivante:

RMF prend la main, après son initialisation, de deux manières: soit pour répondre à une commande qui modifie les options actuellement employées ou soit par des interruptions périodiques précisées grâce à l'option "cycle" qui permettront de donner la main à ces programmes d'observation correspondant aux données que l'on désire.

Quant à RMF, il est la plupart du temps expulsé et ne restent en mémoire réelle (fixés) que les programmes d'observation et le programme qui lui donne la main. Ce n'est que lors des fins d'intervalle que RMF sera réintroduit.

En effet, lorsque RMF est actif, si l'on considère toute la période de l'intervalle, ne sont utiles que les programmes d'observation; c'est pourquoi, on les fixe de telle manière qu'ils puissent être retrouvés lors d'un cycle si RMF est expulsé.

De la même manière que RMF, ces programmes que nous avons écrits et intégrés à RMF possèdent la même structure.

1. un programme d'initialisation.
2. un programme d'observation.

3. un programme de traitement de fin d'intervalle.
4. un programme de rapports.
5. un programme de terminaison.

Il est bien évident que ces programmes s'exécutent également sous les différents programmes de contrôle de RMF.

Regardons de plus près la fonction de chacun de ces programmes.

V.3.1.1. Initialisation.

Le programme d'initialisation a comme fonction de réserver la place nécessaire au programme d'observation pour ranger les données observées.

De plus, initialement, il avait comme but de fixer en mémoire réelle le module d'observation en deux étapes:

La première était de rendre RMF non expulsable pendant que la seconde étape consistait à fixer la zone de mémoire vituelle où "l'observateur" avait été chargé précédemment par ce même programme d'initialisation.

Cependant, cette méthode comportait certaines lacunes, ce qui a nécessité l'emploi d'une autre solution, celle de la "liste fixe". La "liste fixe" est une liste de module conservée par le superviseur et dont tous les membres sont fixés en mémoire virtuelle. Sa gestion est donc entièrement prise en charge par le superviseur et décharge le programme de ses contraintes.

Le programme d'initialisation s'exécute une fois lorsqu'on démarre RMF avec comme option "utilisateur".

Enfin, il réserve, dans une zone du superviseur, des zones de travail pour "l'observateur", zone qu'il

initialise également.

V.3.1.2. Observations.

Le programme de contrôle de RMF va lui passer la main après que lui-même l'ait reçue lors d'une interruption correspondant à un cycle.

L'"observateur" est chargé d'aller recueillir les données sur l'état des tampons VTAM.

La fonction plus détaillée de ce programme est d'aller chercher les valeurs désirées, de mettre à jour les valeurs minimum et maximum correspondantes et d'additionner ce nombre dans un compteur de manière à pouvoir, par après, calculer la moyenne en connaissant le nombre d'observations.

Ce programme, comme le précédent, tourne en clé 0 et de plus ininterruptible, ce qui veut dire qu'un SVC, faisant partie de la codification, provoquera une interruption fatale.

C'est ce module dont nous avons parlé auparavant qui est fixé grâce à la liste fixe.

Il est bien évident que tous ces programmes sont écrits en assembler.

V.3.1.3. Fin d'intervalle.

Ce programme prend la main à la fin de la durée de l'intervalle spécifiée dans les options données à RMF.

De plus, il s'exécute une fois dans la phase d'initialisation.

Dans notre cas, aucune fonction bien précise n'était à exécuter.

Son objectif est de réserver une zone de travail dans laquelle il placera les données que l'observateur lui passera.

Ces données subiront certaines transformations à des fins d'impression. Cette opération aurait d'ailleurs pu avoir lieu dans la phase d'impression.

Enfin, il va réinitialiser la zone de travail de l'observateur qui, rappelons-le, se trouve dans des zones réservées; c'est pourquoi, lui aussi tourne en clé 0.

V.3.1.4. Le programme d'impression.

La fonction du programme d'impression est d'achever la transformation des données et de faire le formattage de manière à pouvoir les imprimer.

En outre, ce programme libèrera les zones réservées par le programme de traitement de fin d'intervalles après avoir, bien sûr, imprimé toutes les données nécessaires, à savoir, pour chaque type de tampon: le minimum, le maximum et la moyenne du nombre de tapons disponibles, du nombre maximum de tampons utilisés et la longueur et le nombre de tampons pour chaque type qui, eux, sont constants sur une session d'observation.

V.3.2. Sous le contrôle de la session 2.

Il reste maintenant à exposer les programmes inclus à la session 2 mais qui, eux, ne donnent que des instantanés de la situation et non des valeurs représentatives d'une période de temps.

Ces programmes sont au nombre de deux:

1/ L'observateur: son but est d'aller chercher les données nécessaires dans les blocs de contrôle de VTAM et de mettre les valeur observées dans une zone réservée à cet effet et dont l'adresse est donnée par le système.

Mais avant que ces programmes puissent s'exécuter, il a fallu recompiler le"menu", qui est la liste des options disponibles en TSO, de manière à ajouter l'option "VTAM".

Ceci étant fait et l'utilisateur ayant choisi l'option VTAM sur son écran, l'observateur va s'exécuter et aller chercher toutes les données, pour les mettre dans la zone réservée par le système et dont l'adresse lui a été passée grâce à un registre. . Les programmes ne sont pas réentrants et tournent en clé utilisateur.

Sa fonction pourrait être comparée à celles de l'observateur et du programme de fin d'intervalle de la session de type 1.

2/ Le rapport: à partir de données dont l'adresse lui est passée, il construit les zones d'impression par la transformation des données hexadécimales et la mise en page des titres.

Cependant, un problème se pose au niveau de l'"impression" sur terminal.

L'adresse de la routine qui se chargera de l'écriture ainsi que l'adresse du bloc contrôle correspondant lui sont passées par RMF.

Ceci va permettre l'impression au terminal adéquat.

Voilà donc en gros les différentes fonctions de ces programmes qui sont écrits, rappelons-le, en assembler.

oooooooooooooo

C O N C L U S I O N S

==========

Le problème qui consiste à définir les paramètres qui fixeront les tampons de VTAM est assez complexe.

Notre objectif était de prendre la mémoire minimum nécessaire.

Dans une première étape, les grandes lignes de VTAM et de l'utilisation de ses tampons ont été exposés.

On a constaté ensuite l'inefficience des méthodes de détermination existantes.

Ceci nous a amené à proposer une méthode susceptible de déterminer ces paramètres.

A cette fin, nous avons construit un simulateur d'allocation et de libération des tampons.

Ce simulateur se servait en partie de données fournies par le système et jusqu'ici inexploitées. Ces données comportaient notamment la longueur ainsi que le moment d'arrivée des messages arrivant dans le système.

Dans une étape préalable à l'exécution définitive de ce simulateur, nous avons déterminé les paramètres P_{x_i}, nombre de tampons de longueur x_i libérés par seconde.

Cette détermination a été rendue possible grâce à la construction d'un outil nous donnant l'état exact des tampons VTAM.

Cet outil a été inclus dans un outil de mesure existant, RMF.

Sa fonction est de collecter des renseignements sur une composante du système: le gérant des ressources.

C'est pourquoi, nous avons tenu à exposer les

grandes lignes de RMF et du gérant des ressources.

Chacun de ces paramètres P_{x_i} a été calculé de manière à faire correspondre le maximum d'octets utilisés pour une longueur de tampon lors de la simulation avec le maximum d'octets utilisés si cette longueur avait été celle de l'installation.

Ceci nous a permis de dégager les besoins maximum propres à chaque longueur de tampon.

On s'est rendu compte ensuite de la nécessité de la présence de plusieurs sessions d'observation reprenant les moments de "pointe" susceptibles de créer les problèmes ultérieurs.

A ce sujet, il faut souligner la dépendance étroite entre la validité des résultats déduits des sorties du simulateur et la représentativité de ces sessions.

L'inclusion de plusieurs sessions devait permettre la détermination de la place mémoire minimum.

A chacune de ces sessions, et pour toute longueur de tampon possible étaient associés les besoins et les caractéristiques correspondants.

Nous avons alors dégagé la place mémoire nécessaire minimum à partir de ces dernières données ainsi que le BSZ adéquat.

La solution idéale à ces problèmes résiderait dans une allocation dynamique qui supprimerait le côté taille mémoire, avec un BSZ minimisant la place prise en moyenne par seconde.

Cependant, dans une installation où l'importance de la ressource mémoire est significative, nous pensons avoir apporté une solution qui permettra de définir des exigences en mémoire moins importante grâce à :

- une meilleure utilisation de la mémoire.
- une connaissance plus exacte des besoins réels.

00000

BIBLIOGRAPHIE

OS/VS2-2	Concept and philosophy	
OS/VS2-2	Programmers guide	
OS/VS2-2	FTS	Advisor Guide Tape Supervisor services
OS/VS2-2	Storage estimates	
RMF	ressource measurement facility	
RMF	program logic manual	
GTF	trace facility	
VTAM	Program logic manual	
VTAM	programmers guide	
VTAM	lògic	
IBM	system journal	
Revue	"Point de l'information"	
Temps réel	J. BAILLY	

Ainsi que certaines brochures internes:

" Some problems of performances"

"VTAM storage requirements"

oooooo

Programme Cobol de
- simulation
- enregistrement des données

```
START                                                                    MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  FLAGS
    8         IDENTIFICATION DIVISION.                                 00010008  *
    8         PROGRAM-ID. COBVTAM.                                     00020009  *
    8         AUTHOR. PONCELET.                                        00030008  *
    8         ENVIRONMENT DIVISION.                                    00040008  *
    8         CONFIGURATION SECTION.                                   00050008  *
    8         SOURCE-COMPUTER.  IBM-370.                               00060008  *
    8         OBJECT-COMPUTER.  IBM-370.                               00070008  *
    8         SPECIAL-NAMES.                                           00080054  *
   12             DECIMAL-POINT IS COMMA.                             00090054  *
    8         INPUT-OUTPUT SECTION.                                    00100008  *
    8         FILE-CONTROL.                                            00110012  *
   12             SELECT FPRIM ASSIGN TO UT-S-PRINT.                  00120015  *
    8         DATA DIVISION.                                           00130000
    8         FILE SECTION.                                            00140008  *
    8         FD   FPRIM      LABEL RECORD IS OMITTED.                 00150015  *
    8         01   ZIMP PIC X(132).                                    00160021  *
    8         WORKING-STORAGE SECTION.                                 00170061  *
    8         01   ZIMP1.                                              00180000
   12              02 CHART  PIC 9.                                    00190053  *
   12              02 LIM1 PIC 9(4).                                   00200054  *
   12              02 FILLER PIC X(1).                                 00210049  *
   12              02 AST0   PIC X.                                    00220054  *
   12              02 FILLER PIC X(1).                                 00230049  *
   12              02 NBRE  PIC ZZZ.ZZZ.                               00240065  *
   12              02 FILLER PIC X(1).                                 00250049  *
   12              02 AST5   PIC X.                                    00260054  *
   12              02 FILLER PIC X(1).                                 00270049  *
   12              02 NBIN   PIC ZZ.ZZZ.                               00280061  *
   12              02 FILLER PIC X(1).                                 00290049  *
   12              02 AST3   PIC X.                                    00300054  *
   12              02 FILLER PIC X(1).                                 00310049  *
   12              02 NBOUT  PIC ZZ.ZZZ.                               00320061  *
   12              02 FILLER PIC X(1).                                 00330049  *
   12              02 AST2   PIC X.                                    00340054  *
   12              02 FILLER PIC X(1).                                 00350049  *
   12              02 TAUX   PIC    9,99.                              00360067  *
   12              02 FILLER PIC X(1).                                 00370049  *
   12              02 AST1   PIC X.                                    00380054  *
   12              02 FILLER PIC X(1).                                 00390049  *
   12              02 FREQ   PIC    9,99.                              00400067  *
    8         01   ZIMP2.                                             00410005  *
   12              02 CHAR3  PIC 9.                                    00420053  *
   12              02 NOMAL PIC X(6).                                  00430053  *
   12              02 FILLER PIC X(1).                                 00440049  *
   12              02 AST2Z  PIC X.                                    00450054  *
   12              02 FILLER PIC X(1).                                 00460049  *
   12              02 NUMB   PIC ZZZ.ZZZ.                              00470065  *
   12              02 FILLER PIC X(1).                                 00480049  *
   12              02 AST2E  PIC X.                                    00490054  *
   12              02 FILLER PIC X(1).                                 00500049  *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12              02 LONGA  PIC ZZZ.ZZZ.                                          00510065   *
   12              02 FILLER PIC X(1).                                             00520049   *
   12              02 AST2R  PIC X.                                                00530054   *
   12              02 FILLER PIC X(1).                                             00540049   *
   12              02 NBINAPL PIC ZZZ.ZZZ.                                         00550065   *
   12              02 FILLER PIC X(1).                                             00560049   *
   12              02 AST6   PIC X.                                                00570054   *
   12              02 FILLER PIC X(1).                                             00580049   *
   12              02 NBOUTAPL PIC ZZZZZZ.                                         00590065   *
   12              02 FILLER PIC X(1).                                             00600049   *
   12              02 AST8   PIC X.                                                00610054   *
   12              02 FILLER PIC X(1).                                             00620049   *
   12              02 TAUXA  PIC 9,99.                                             00630067   *
   12              02 FILLER PIC X(1).                                             00640049   *
   12              02 ASTA   PIC X.                                                00650054   *
   12              02 FILLER PIC X(1).                                             00660049   *
   12              02 FREQA  PIC 9,99.                                             00670067   *
    8          01  ZIMP3.                                                         00680005   *
   12              02 FGHJ PIC 9.                                                  00690053   *
   12              02 DAYT PIC ZZZ.                                                00700054   *
   12              02 FILLER PIC X(1).                                             00710049   *
   12              02 ASTS   PIC X.                                                00720054   *
   12              02 FILLER PIC X(1).                                             00730049   *
   12              02 HOURT PIC ZZ.                                                00740054   *
   12              02 FILLER PIC X(1).                                             00750049   *
   12              02 ASTD   PIC X.                                                00760054   *
   12              02 FILLER PIC X(1).                                             00770049   *
   12              02 MINIT PIC ZZ.                                                00780054   *
   12              02 FILLER PIC X(1).                                             00790049   *
   12              02 ASTF   PIC X.                                                00800054   *
   12              02 FILLER PIC X(1).                                             00810049   *
   12              02 SECOND PIC ZZ.                                               00820054   *
    8          01  ZIMP4.                                                         00830005   *
   12              02 JHNG PIC 9.                                                  00840053   *
   12              02 NBREINT PIC   ZZ.ZZZ.                                        00850067   *
   12              02 FILLER PIC X(1).                                             00860049   *
   12              02 ASTG   PIC X.                                                00870054   *
   12              02 FILLER PIC X(1).                                             00880049   *
   12              02 LENGINT PIC    ZZZZ.                                         00890067   *
   12              02 FILLER PIC X(1).                                             00900049   *
   12              02 ASTH   PIC X.                                                00910054   *
   12              02 FILLER PIC X(1).                                             00920049   *
   12              02 TAUXINT PIC    9,99.                                         00930067   *
   12              02 FILLER PIC X(1).                                             00940049   *
   12              02 ASTJ   PIC X.                                                00950054   *
   12              02 FILLER PIC X(1).                                             00960049   *
   12              02 TEMPSEC PIC  ZZZZ.                                           00970067   *
    8          01  ZIMP5.                                                         00980005   *
   12              02 SBVJ PIC 9.                                                  00990053   *
   12              02 TIT PIC X(32).                                               01000070   *
```

```
START                                                                                                    MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8                   FLAGS
   12             02 TITZ PIC X(40).                                                                    01010070   *
    8          01   TRAV3 PIC 9(18) USAGE COMP-3 VALUE 0.                                              01020068   *
    8          01   ZIMP10.                                                                            01030066   *
   12             02 TRY OCCURS 8 INDEXED BY I8.                                                       01040066   *
   12             03 PAGEDS PIC ZZZZ.                                                                  01050076   *
    8          01   ZIMP9.                                                                             01060005   *
   12             02 SPEMS PIC 9.                                                                      01070053   *
   12             02 ZVAL PIC  ZZZ.                                                                    01080067   *
   12             02 FILLER PIC X(1).                                                                  01090049   *
   12             02 ASTL   PIC X.                                                                     01100054   *
   12             02 FILLER PIC X(1).                                                                  01110049   *
   12             02 ZACTUEL   PIC    ZZZZZ.                                                           01120067   *
   12             02 FILLER PIC X(1).                                                                  01130049   *
   12             02 ASTW   PIC X.                                                                     01140054   *
   12             02 FILLER PIC X(1).                                                                  01150049   *
   12             02 ZMAXINT   PIC ZZZZZZZ.                                                            01160067   *
   12             02 FILLER PIC X(1).                                                                  01170049   *
   12             02 ASTX   PIC X.                                                                     01180054   *
   12             02 FILLER PIC X(1).                                                                  01190049   *
   12             02 MOYE      PIC   99,99.                                                            01200067   *
   12             02 FILLER PIC X(1).                                                                  01210049   *
   12             02 ASTV   PIC X.                                                                     01220054   *
   12             02 FILLER PIC X(1).                                                                  01230049   *
   12             02 MBYTE     PIC    Z.ZZZ.                                                           01240067   *
   12             02 FILLER PIC X(1).                                                                  01250049   *
   12             02 ASTB   PIC X.                                                                     01260054   *
   12             02 FILLER PIC X(1).                                                                  01270049   *
   12             02 TBYTE     PIC ZZZZZZZ.                                                            01280070   *
    8          01   CPTL  PIC 9(6) USAGE COMP-3 VALUE 0.                                               01290065   *
    8          01   IT PIC 9(6) VALUE 0.                                                               01300065   *
    8          01   DYNAM.                                                                             01310005   *
   12             02 BUFF      OCCURS 18 INDEXED BY I6.                                               01320031   *
   13              03 VALEUR   PIC 9(4) USAGE COMP-3.                                                  01330067   *
   13              03 TOTB     PIC 9(8) USAGE COMP-3.                                                  01340067   *
   13              03 TOTALI   PIC 9(8) USAGE COMP-3.                                                  01350067   *
   13              03 ACTUEL   PIC 9(8) USAGE COMP-3.                                                  01360067   *
   13              03 TOTT     PIC 9(8) USAGE COMP-3.                                                  01370067   *
   13              03 MAXINT   PIC 9(8) USAGE COMP-3.                                                  01380067   *
   13              03 MAXTOT   PIC 9(8) USAGE COMP-3.                                                  01390067   *
   13              03 PAGEI    OCCURS  8 INDEXED BY I7.                                                01400031   *
   16                04 BTHPI PIC 9(4) USAGE COMP-3.                                                   01410022   *
    9          01   PXUN.                                                                              01420005   *
   13              02 PPL0   PIC  9(4) VALUE 524 USAGE COMP-3.                                         01430084   *
   13              02 PPL1   PIC  9(4) VALUE 514  USAGE COMP-3.                                        01440085   *
   13              02 PPL2   PIC  9(4) VALUE 496 USAGE COMP-3.                                         01450084   *
   13              02 PPL3   PIC  9(4) VALUE 481 USAGE COMP-3.                                         01460084   *
   13              02 PPL4   PIC  9(4) VALUE 472 USAGE COMP-3.                                         01470084   *
   13              02 PPL5   PIC  9(4) VALUE 463 USAGE COMP-3.                                         01480086   **
   13              02 PPL6   PIC  9(4) VALUE 455 USAGE COMP-3.                                         01490086   **
   13              02 PPL7   PIC  9(4) VALUE 450 USAGE COMP-3.                                         01500084   *
```

```
START                                                                                                    MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8                    FLAGS
  13                02  PPL8    PIC  9(4) VALUE 446 USAGE COMP-3.                                      01510084   *
  13                02  PPL9    PIC  9(4) VALUE 442 USAGE COMP-3.                                      01520084   *
  13                02  PPLA    PIC  9(4) VALUE 438 USAGE COMP-3.                                      01530084   *
  13                02  PPLZ    PIC  9(4) VALUE 435 USAGE COMP-3.                                      01540084   *
  13                02  PPLE    PIC  9(4) VALUE 431 USAGE COMP-3.                                      01550084   *
  13                02  PPLR    PIC  9(4) VALUE 429 USAGE COMP-3.                                      01560084   *
  13                02  PPL1    PIC  9(4) VALUE 427 USAGE COMP-3.                                      01570084   *
  13                02  PPT1    PIC  9(4) VALUE 424 USAGE COMP-3.                                      01580084   *
  13                02  PPY1    PIC  9(4) VALUE 423 USAGE COMP-3.                                      01590084   *
  13                02  PPI1    PIC  9(4) VALUE 422 USAGE COMP-3.                                      01600084   *
   8          01   PXRED REDEFINES PXUN.                                                              01610011   *
  13                02  OKLIOU  OCCURS 18 INDEXED BY I5.                                              01620031   *
  14                  03 PX    PIC   9(4) USAGE COMP-3.                                              01630022   *
   8          01   PLACE       PIC   9(18) VALUE 0 USAGE COMP-3.                                     01640061   *
   8          01   TEMP        PIC   9(18) VALUE 0 USAGE COMP-3.                                     01650025   *
   8          01   LONG        PIC   9(4) VALUE 0 USAGE COMP-3.                                      01660023   *
   8          01   NEWTIME     PIC   9(18)  USAGE COMP-3 VALUE 0.                                    01670025   *
   8          01   OLDTIME     PIC   9(18)  USAGE COMP-3 VALUE 0.                                    01680025   *
   8          01   TEMPINT     PIC  9(18) USAGE COMP-3 VALUE 0.                                      01690052   *
   8          01   PAGE1      PIC  9(18) USAGE COMP-3 VALUE 0.                                       01700061   *
   8          01   PAGE2      PIC  9(18) USAGE COMP-3 VALUE 0.                                       01710061   *
   8          01   PAGE3      PIC  9(18) USAGE COMP-3 VALUE 0.                                       01720061   *
   8          01   TRAV1       PIC 9(18)  USAGE COMP-3 VALUE 0.                                      01730061   *
   8          01   TRAV2       PIC  9(18) USAGE COMP-3 VALUE 0.                                      01740061   *
   8          01   COMPLACE    PIC   9(18) VALUE 0 USAGE COMP-3.                                     01750065   *
   8          01   NBUFFER     PIC  9(18) USAGE COMP-3 VALUE 0.                                      01760065   *
   8          01   OPTIMUM     PIC  9(18) USAGE COMP-3 VALUE 0.                                      01770065   *
   8          01    TIT1 PIC X(19) VALUE 'DATE  HOUR MIN  SEC'.                                     01780050   *
   8          01    TIT2 PIC X(17) VALUE 'FOR THIS INTERVAL'.                                       01790011   *
   8          01   TIT3 PIC X(32) VALUE ' FOR ALL THE INTERVAL '.                                   01800074   *
   8          01   TIT4 PIC X(32) VALUE ' LONG      NBRE      NBIN    NBOUT'.                       01810070   *
   8          01   TITA PIC X(40) VALUE '  TAUX   FREQ'.                                           01820070   *
   8          01   TIT5 PIC X(42) VALUE ' BY APPLICATION RUNNING '.                                 01830074   *
   8          01   TIT8 PIC X(32) VALUE '     NOM       NBRE      LONG      N'.                    01840070   *
   8          01   TITB PIC X(40) VALUE '          NBOUT   TAUX   FREQ'.                          01850070   *
   8          01    TIT6 PIC X(30) VALUE 'RESUMED OF THE VALUES OBSERVED'.                          01860011   *
   8          01   TIT12 PIC X(30) VALUE ' NBRE     LONG   TAUX   FREQ'.                          01870055   *
   8          01   TIT10 PIC X(42) VALUE ' NUMBER PAGES XI GOOD '.                                  01880074   *
   8          01   TIT9 PIC X(32) VALUE '  BSZ   USED   MAX     E(USED)    '.                     01890070   *
   8          01   TITC PIC X(32) VALUE ' A(L)     TOT'.                                           01900070   *
   8          01   TIT14 PIC X(130) VALUE SPACES.                                                  01910070   *
   8          01   TIT20 PIC X(70) VALUE '**********************************'.                     01920070   *
   8          01   LIEN.                                                                           01930029   *
  12                02 IN-OUT PIC 9 VALUE 0.                                                       01940029   *
  12                02 APPL  PIC X(8) VALUE SPACES.                                                01950037   *
  12                02 LONG1 PIC 9(4) USAGE COMP-4 VALUE 0.                                        01960029   *
  12                02 MEMT  PIC 9 VALUE 0.                                                        01970029   *
  12                02 TEMP1 PIC 9(16) USAGE COMP-4 VALUE 0.                                       01980029   *
   8          01   BSZ  PIC  9(4) USAGE COMP-3 VALUE 0.                                           01990031   *
   8          01   NUMTOT  PIC  9(18) VALUE 0 USAGE COMP-3.                                       02000065   *
```

VALEUR PASSEES PAR LE PROGRAMME ASSEMBLER

```
START                                                                                                    MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8   FLAGS
    8          01  NUMINT  PIC  9(18) VALUE 0 USAGE COMP-3.                              02010065   *
    8          01  TNUMAPL PIC  9(18) VALUE 0 USAGE COMP-4.                              02020065   *
    8          01  LONGINT PIC  9(18) VALUE 0 USAGE COMP-3.                              02030065   *
    8          01  LONGTOT PIC  9(18) VALUE 0 USAGE COMP-3.                              02040065   *
    8          01  SECI    PIC  9(18) VALUE 0 USAGE COMP-3.                              02050065   *
    8          01  SECT    PIC  9(18) VALUE 0 USAGE COMP-3.                              02060065   *
    8          01  INTERVAL.                                                             02070005   *
   12              02 ICLASSE OCCURS 301 INDEXED BY I1.                                  02080031   *
   13                03 INUMMESS    PIC 9(8) USAGE COMP-3.                               02090065   *
   13                03 ILONGMOY  PIC 9(9) USAGE COMP-3.                                 02100065   *
   13                03 INUMIN     PIC 9(8) USAGE COMP-3.                               02110065   *
   13                03 INUMOUT    PIC 9(8) USAGE COMP-3.                               02120065   *
   13                03 TNUMMESS   PIC 9(8) USAGE COMP-3.                               02130065   *
   13                03 TLONGMOY   PIC 9(9) USAGE COMP-3.                               02140065   *
   13                03 TNUMIN     PIC 9(9) USAGE COMP-3.                               02150065   *
   13                03 TNUMOUT    PIC 9(9) USAGE COMP-3.                               02160065   *
    8          01  IAPPLIC.                                                              02170005   *
   12              02 IAPPL OCCURS 50 INDEXED BY I3.                                     02180031   *
   13                03 INOM PIC X(8).                                                  02190019   *
   13                03 IMESSAPL   PIC 9(4) USAGE COMP-3.                               02200067   *
   13                03 ILONGAPL   PIC 9(8) USAGE COMP-3.                               02210067   *
   13                03 IAPLIN     PIC 9(4) USAGE COMP-3.                               02220067   *
   13                03 IAPLOUT    PIC 9(4) USAGE COMP-3.                               02230067   *
   13                03 TMESSAPL   PIC 9(4) USAGE COMP-3.                               02240067   *
   13                03 TLONGAPL   PIC 9(8) USAGE COMP-3.                               02250067   *
   13                03 TAPLIN     PIC 9(4) USAGE COMP-3.                               02260067   *
   13                03 TAPLOUT    PIC 9(4) USAGE COMP-3.                               02270067   *
    8          01  LASTTIME PIC 9(18) VALUE 0 USAGE COMP-3.                             02280025   *
    8          01  SAVETIME PIC 9(18) VALUE 0 USAGE COMP-3.                             02290025   *
    8          PROCEDURE DIVISION.                                                     02300009   *
    8          CALC.                                                                   02310009   *
   12              SET I6 TO 1.                                                        02320031   *
   12              MOVE 80 TO BSZ.                                                     02330069   *
    8          CALC0.                                                                  02340011   *
   12              MOVE 10688 TO PAGE1.                                                02350076   *
   12              MOVE BSZ TO VALEUR (I6).                                            02360031   *
   12              MOVE 0 TO TOTB (I6).                                                02370031   *
   12              MOVE 0 TO TOTALI (I6).                                              02380031   *
   12              MOVE 0 TO ACTUEL (I6).                                              02390031   *
   12              MOVE 0 TO TOTT (I6).                                                02400031   *
   12              MOVE 0 TO MAXINT (I6).                                              02410031   *
   12              MOVE 0 TO MAXTOT (I6).                                              02420031   *
   12              SET I7 TO 1.                                                        02430031   *
    8          CALC1.                                                                  02440009   *
   12              MOVE BSZ TO PAGE2.                                                  02450009   *
   12              ADD 72 TO PAGE2.                                                    02460022   *
   12              DIVIDE PAGE1 BY PAGE2 GIVING PAGE3.                                 02470009   *
   12              ADD 1 TO PAGE3.                                                     02480009   *
   12              MOVE PAGE3 TO BTHPI (I6, I7).                                       02490031   *
   12              SET I7 UP BY 1.                                                     02500031   *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12             ADD  4096 TO PAGE1.                                                  02510009 *
   12             IF I7 LESS THAN 9 GO TO CALC1.                                       02520031 *
   12             SET I6 UP BY 1. ADD 8 TO BSZ.                                        02530031 *
   12             IF I6 LESS THAN 19 GO TO CALC0.                                      02540031 *
   12             SET I1 TO 1.                                                         02550040 *
    8         RECY. MOVE 0 TO INUMMESS (I1).                                          02560037 *
   12             MOVE 0 TO TNUMMESS (I1).                                             02570040 *
   12             MOVE 0 TO ILONGMOY (I1).                                             02580037 *
   12             MOVE 0 TO TLONGMOY (I1).                                             02590040 *
   12             MOVE 0 TO INUMIN (I1).                                               02600037 *
   12             MOVE 0 TO TNUMIN (I1).                                               02610040 *
   12             MOVE 0 TO INUMOUT (I1).                                              02620037 *
   12             MOVE 0 TO TNUMOUT (I1).                                              02630040 *
    8         RECZ. SET I1 UP BY 1.                                                   02640040 *
   12             IF I1 LESS THAN 302 GO TO RECY.                                      02650037 *
   12             SET I3 TO 1.                                                         02660040 *
    8         AA. MOVE 0 TO IMESSAPL (I3).                                            02670037 *
   12             MOVE 0 TO TMESSAPL (I3).                                             02680040 *
   12             MOVE 0 TO ILONGAPL (I3).                                             02690037 *
   12             MOVE 0 TO TLONGAPL (I3).                                             02700040 *
   12             MOVE 0 TO IAPLIN (I3).                                               02710037 *
   12             MOVE 0 TO TAPLIN (I3).                                               02720040 *
   12             MOVE 0 TO IAPLOUT (I3).                                              02730037 *
   12             MOVE 0 TO TAPLOUT (I3).                                              02740040 *
    8         R8. SET I3 UP BY 1.                                                     02750040 *
    8         HH. IF I3 LESS THAN 51 GO TO AA.                                        02760037 *
   12             OPEN OUTPUT FPRIM.                                                   02770011 *
    8         READVAR.                                                                02780009 *
   12             CALL 'ASMREAD' USING LIEN.                                           02790019 *
   12             MOVE TEMP1 TO TEMP.                                                  02800023 *
   12             IF MEMT = 8 GO TO FIN.                                               02810019 *
   12             IF MEMT NOT = 0 GO TO ENDINT.                                        02820009 *
    8         RECNORM.                                                                02830009 *
   12             MOVE LONG1 TO  LONG.                                                 02840040 *
    8         COMP1.                                                                  02850009 *
   12             IF LONG GREATER THAN 301 SET I1 TO 301 GO TO SUIT1.                  02860040 *
   12             SET I1 TO LONG1.                                                     02870040 *
    8         SUIT1.                                                                  02880009 *
   12             ADD 1 TO INUMMESS (I1).                                              02890031 *
   12             ADD LONG TO ILONGMOY (I1).                                           02900031 *
   12             IF IN-OUT = 1 GO TO INBOUND.                                         02910009 *
   12             ADD 1 TO INUMOUT (I1). GO TO SUIT2.                                  02920031 *
    8         INBOUND.                                                                02930009 *
   12             ADD 1 TO INUMIN (I1).                                                02940031 *
    8         SUIT2.                                                                  02950009 *
   12             IF TNUMAPL = 0 GO TO DEBAPL.                                         02960040 *
   12             SET I3 TO 1.                                                         02970040 *
    8         SUIT3.                                                                  02980009 *
   12             IF INOM (I3) = APPL GO TO UPDTAPL.                                   02990031 *
   12             IF I3  EQUAL TO TNUMAPL GO TO DEBAPL.                                03000040 *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12            SET I3 UP BY 1. GO TO SUIT3.                                          03010031  *
    8       DEBAPL.                                                                    03020011  *
   12            ADD 1 TO TNUMAPL.                                                     03030040  *
   12            SET I3 TO TNUMAPL.                                                    03040041  *
   12            MOVE APPL TO INOM (I3).                                               03050031  *
    8       UPDTAPL.                                                                   03060009  *
   12            ADD 1 TO IMESSAPL (I3).                                               03070031  *
   12            ADD LONG TO ILONGAPL (I3).                                            03080031  *
   12            IF IN-OUT = 2 GO TO OUTBOUND.                                         03090009  *
   12            ADD 1 TO IAPLIN (I3).                                                 03100031  *
   12            GO TO SUIT4.                                                          03110009  *
    8       OUTBOUND.                                                                  03120009  *
   12            ADD 1 TO IAPLOUT (I3).                                                03130031  *
    8       SUIT4.                                                                     03140009  *
   12            ADD 1 TO NUMINT.                                                      03150014  *
   12            ADD LONG TO LONGINT.                                                  03160009  *
   12            SUBTRACT OLDTIME FROM TEMP GIVING TEMPINT.                            03170055  *
   12            MOVE TEMP TO OLDTIME.                                                 03180009  *
   12            DIVIDE TEMPINT BY 625 GIVING TEMPINT.                                 03190037  *
   12            SET I6 TO 1. SET I5 TO 1.                                             03200031  *
    8       FOLO.                                                                      03210015  *
   12            DIVIDE LONG  BY VALEUR (I6) GIVING TRAV1.                             03220031  *
   12            MULTIPLY TRAV1 BY VALEUR (I6) GIVING TRAV2.                           03230066  *
   12            IF TRAV2 NOT LESS THAN LONG GO TO K5.                                 03240066  *
   12            ADD 1 TO TRAV1.                                                       03250009  *
    8       K5. ADD TRAV1 TO TOTB (I6).                                                03260066  *
   12            MULTIPLY TRAV1 BY 10000 GIVING TRAV1.                                 03270029  *
   12            ADD TRAV1 TO ACTUEL (I6).                                             03280031  *
   12            MOVE PX (I5) TO TRAV2.                                                03290031  *
   12            MULTIPLY TRAV2 BY TEMPINT GIVING TRAV2.                               03300061  *
   12            SUBTRACT TRAV2 FROM ACTUEL (I6).                                      03310031  *
   12            IF ACTUEL (I6) LESS THAN MAXINT (I6) GO TO FOL1.                      03320031  *
   12            MOVE ACTUEL (I6) TO MAXINT (I6).                                      03330031  *
    8       FOL1.                                                                      03340060  *
   12            SET I6 UP BY 1. SET I5 UP BY 1.                                       03350031  *
   12            IF I6 LESS THAN 19 GO TO FOLO.                                        03360032  *
   12            GO TO READVAR.                                                        03370009  *
    8       ENDINT. ADD NUMINT TO NUMTOT. IF MEMT = 2 GO TO TOTA.                      03380050  *
   12            IF NUMTOT  = 0 GO TO MEP.                                             03390009  *
   12            SET I1 TO 1.                                                          03400040  *
    8       CONTIN.                                                                    03410009  *
   12            ADD INUMMESS (I1) TO TNUMMESS (I1).                                   03420040  *
   12            ADD ILONGMOY (I1) TO TLONGMOY (I1).                                   03430040  *
   12            ADD INUMIN (I1) TO TNUMIN (I1).                                       03440040  *
   12            ADD INUMOUT (I1) TO TNUMOUT (I1).                                     03450040  *
   12            SET I1 UP BY 1.                                                       03460040  *
   12            IF I1 LESS THAN 302 GO TO CONTIN.                                     03470031  *
   12            ADD LONGINT TO LONGTOT.                                               03480041  *
   12            SET I3 TO 1.                                                          03490043  *
    8       USERED.                                                                    03500009  *
```

CALCUL

Si LORS DE

L'ARRIVEE D'UN

MESSAGE.

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12              ADD IMESSAPL (I3) TO TMESSAPL (I3).                                 03510040  *
   12              ADD ILONGAPL (I3) TO TLONGAPL (I3).                                 03520040  *
   12              ADD IAPLIN (I3) TO TAPLIN (I3).                                     03530040  *
   12              ADD IAPLOUT (I3) TO TAPLOUT (I3).                                   03540040  *
   12              SET I3 UP BY 1.                                                     03550031  *
   12              IF I3 NOT GREATER THAN TNUMAPL GO TO USERED.                        03560043  *
   12              SET I6 TO 1.                                                        03570054  *
    8          W0. ADD TOTB (I6) TO TOTT (I6).                                         03580054  *
   12              SET I6 UP BY 1.                                                     03590054  *
   12              IF I6 LESS THAN 19 GO TO W0.                                        03600054  *
   12              GO TO MEP.                                                          03610049  *
    8          TOTA.                                                                   03620009  *
   12              SET I1 TO 1.                                                        03630040  *
   12              SET I3 TO 1.                                                        03640040  *
   12              SET I6 TO 1.                                                        03650031  *
    8          PTRE. IF TNUMMESS (I1) = 0 GO TO UY.                                    03660040  *
   12              MOVE TNUMMESS (I1) TO INUMMESS (I1).                                03670040  *
   12              MOVE TLONGMOY (I1) TO ILONGMOY (I1).                                03680040  *
   12              MOVE TNUMIN (I1) TO INUMIN (I1).                                    03690040  *
   12              MOVE TNUMOUT (I1) TO INUMOUT (I1).                                  03700040  *
    8          UY. SET I1 UP BY 1.                                                     03710040  *
   12              IF I1 LESS THAN 302 GO TO PTRE.                                     03720031  *
    8          MM. MOVE TOTT (I6) TO TOTB (I6).                                        03730031  *
   12              MOVE MAXTOT (I6) TO MAXINT (I6).                                    03740033  *
   12              SET I6 UP BY 1.                                                     03750040  *
   12              IF I6 LESS THAN 19 GO TO MM.                                        03760033  *
    8          S8. MOVE TLONGAPL (I3) TO ILONGAPL (I3).                                03770040  *
   12              MOVE TMESSAPL (I3) TO IMESSAPL (I3).                                03780040  *
   12              MOVE TAPLIN (I3) TO IAPLIN (I3).                                    03790040  *
   12              MOVE TAPLOUT (I3) TO IAPLOUT (I3).                                  03800040  *
   12              SET I3 UP BY 1.                                                     03810040  *
   12              IF I3 NOT GREATER THAN TNUMAPL GO TO S8.                            03820044  *
   12              MOVE NUMTOT TO NUMINT.                                              03830009  *
   12              MOVE SECT TO SECI.                                                  03840054  *
   12              MOVE LONGTOT TO LONGINT.                                            03850009  *
   12              WRITE ZIMP FROM TIT3.                                               03860070  *
   12              GO TO SUIT9.                                                        03870009  *
    8          MEP.                                                                    03880009  *
   12              WRITE ZIMP FROM TIT14. WRITE ZIMP FROM TIT20.                       03890070  *
   12              WRITE ZIMP FROM TIT14.                                              03900070  *
   12              WRITE ZIMP FROM TIT1.                                               03910070  *
   12              IF NUMTOT = 0 MOVE TEMP TO OLDTIME.                                 03920029  *
   12              DIVIDE TEMP BY 62500 GIVING SAVETIME.                               03930023  *
   12              MOVE SAVETIME TO TEMP.                                              03940015  *
   12              SUBTRACT 2458166400 FROM TEMP.                                      03950009  *
   12              MOVE 0 TO IT.                                                       03960054  *
    8          BOUC4.                                                                  03970009  *
   12              IF TEMP LESS THAN 86400 GO TO BOUC5.                                03980009  *
   12              SUBTRACT 86400 FROM TEMP.                                           03990042  *
   12              ADD 1 TO IT.                                                        04000031  *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12              GO TO BOUC4.                                                        04010009   *
    8         BOUC5. SUBTRACT 30 FROM IT.                                             04020053   *
   12              MOVE IT TO DAYT.                                                    04030031   *
   12              MOVE 0 TO IT.                                                       04040054   *
    8         BOUC6.                                                                  04050009   *
   12              IF TEMP LESS THAN 3600 GO TO BOUC7.                                 04060009   *
   12              SUBTRACT 3600 FROM TEMP.                                            04070009   *
   12              ADD 1 TO IT.                                                        04080031   *
   12              GO TO BOUC6.                                                        04090009   *
    8         BOUC7.                                                                  04100009   *
   12              MOVE IT TO HOURT.                                                   04110031   *
   12              MOVE 0 TO IT.                                                       04120054   *
    8         BOUC8.                                                                  04130009   *
   12              IF TEMP LESS THAN 60 GO TO BOUC9.                                   04140009   *
   12              SUBTRACT 60 FROM TEMP.                                              04150009   *
   12              ADD 1 TO IT.                                                        04160031   *
   12              GO TO BOUC8.                                                        04170009   *
    8         BOUC9.                                                                  04180009   *
   12              MOVE IT TO MINIT. MOVE TEMP TO SECOND.                              04190043   *
   12              MOVE '*' TO ASTS ASTF ASTD.                                         04200054   *
   12              WRITE ZIMP FROM ZIMP3.                                              04210061   *
   12              WRITE ZIMP FROM TIT14. WRITE ZIMP FROM TIT14.                       04220070   *
   12              IF NUMTOT = 0 MOVE SAVETIME TO LASTTIME GO TO READVAR.              04230009   *
    8         S9. SUBTRACT LASTTIME FROM SAVETIME GIVING SECI.                        04240043   *
   12              MOVE SAVETIME TO LASTTIME.                                          04250009   *
   12              ADD SECI TO SECT.                                                   04260009   *
    8         SECIT.                                                                  04270009   *
   12              WRITE ZIMP FROM TIT2.                                               04280071   *
    8         SUIT9. WRITE ZIMP FROM TIT14.                                           04290070   *
   12              WRITE ZIMP FROM TIT14.                                              04300070   *
   12              SET I1 TO 1. MOVE 1 TO CPTL.                                        04310076   *
   12              MOVE TIT4 TO TIT. MOVE TITA TO TITZ. WRITE ZIMP FROM ZIMP5.         04320070   *
    8         IMP1. IF NUMINT = 0 GO TO IF3.                                          04330011   *
   12              IF SECI = 0 GO TO IF3.                                              04340065   *
   12              IF INUMMESS (I1) = 0 GO TO DD.                                      04350031   *
   12              DIVIDE INUMMESS (I1) BY NUMINT GIVING TAUX.                         04360031   *
   12              DIVIDE INUMMESS (I1) BY SECI GIVING FREQ.                           04370031   *
   12              MOVE INUMMESS (I1) TO NBRE.                                         04380031   *
   12              MOVE INUMIN (I1) TO NBIN.                                           04390031   *
   12              MOVE INUMOUT (I1) TO NBOUT.                                         04400031   *
   12              MOVE CPTL TO LIM1.                                                  04410033   *
   12              MOVE '*' TO AST0 AST1 AST2 AST3  AST5.                              04420068   *
   12              WRITE ZIMP FROM ZIMP1.                                              04430061   *
    8         DD. SET I1 UP BY 1. ADD 1 TO CPTL.                                      04440033   *
   12              IF I1 LESS THAN 302 GO TO IMP1.                                     04450031   *
   12              SET I3 TO 1.                                                        04460031   *
   12              WRITE ZIMP FROM TIT14. WRITE ZIMP FROM TIT14.                       04470070   *
   12              WRITE ZIMP FROM TIT5 .                                              04480056   *
   12              MOVE TIT8 TO TIT. MOVE TITB TO TITZ. WRITE ZIMP FROM ZIMP5.         04490070   *
    8         IMP2.                                                                   04500009   *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
   12              IF IMESSAPL (I3) = 0 GO TO II.                                      04510040   *
   12              DIVIDE IMESSAPL (I3) BY NUMINT  GIVING TAUXA.                       04520031   *
   12              DIVIDE IMESSAPL (I3) BY SECI  GIVING FREQA.                         04530053   *
   12              DIVIDE ILONGAPL (I3) BY IMESSAPL (I3) GIVING LONGA.                 04540041   *
   12              MOVE  INOM (I3) TO NOMAL.                                           04550031   *
   12              MOVE IAPLIN (I3) TO NBINAPL.                                        04560031   *
   12              MOVE IAPLOUT (I3) TO NBOUTAPL.                                      04570031   *
   12              MOVE IMESSAPL (I3) TO NUMB.                                         04580031   *
   12              MOVE '*' TO AST2Z AST2E AST2R AST6  AST8  ASTA.                     04590068   *
   12              WRITE ZIMP FROM ZIMP2.                                              04600061   *
    8         II. IF I3 NOT GREATER THAN TNUMAPL SET I3 UP BY 1 GO TO IMP2.           04610075   *
   12              WRITE ZIMP FROM TIT14. WRITE ZIMP FROM TIT14.                       04620070   *
   12              WRITE ZIMP FROM  TIT6. WRITE ZIMP FROM TIT12.                       04630070   *
   12              MOVE NUMINT TO NBREINT.                                             04640009   *
   12              DIVIDE LONGINT BY NUMINT GIVING LENGINT.                            04650029   *
   12              DIVIDE NUMINT BY SECI GIVING TAUXINT.                               04660029   *
   12              MOVE SECI TO TEMPSEC.                                               04670009   *
   12              MOVE '*' TO ASTG ASTH ASTJ.                                         04680054   *
   12              WRITE ZIMP FROM ZIMP4.                                              04690061   *
    8         RESTBUF.                                                                04700009   *
   12              SET I6 TO 1. WRITE ZIMP FROM TIT14. WRITE ZIMP FROM TIT14.          04710070   *
    8         PG. MOVE TIT9 TO TIT. MOVE TITC TO TITZ. WRITE ZIMP FROM ZIMP5.         04720070   *
   12              IF MAXINT (I6) LESS THAN MAXTOT (I6) GO TO DFG.                     04730045   *
   12              MOVE MAXINT (I6) TO MAXTOT (I6).                                    04740031   *
    8         DFG.                                                                    04750009   *
   12              ADD 5000 TO ACTUEL (I6).                                            04760031   *
   12              DIVIDE ACTUEL (I6) BY 10000 GIVING ZACTUEL.                         04770043   *
   12              ADD 5000 TO MAXINT (I6).                                            04780031   *
   12              DIVIDE MAXINT (I6) BY 10000 GIVING TRAV1.                           04790045   *
   12              MOVE VALEUR (I6) TO TOTALI (I6).                                    04800070   *
   12              ADD 72 TO TOTALI (I6).                                              04810070   *
   12              MULTIPLY TOTALI (I6) BY TOTB (I6) GIVING TOTALI (I6).               04820070   *
   12              MOVE TOTALI (I6) TO TBYTE.                                          04830031   *
    8         COM3. MOVE VALEUR (I6) TO TRAV3. ADD 72 TO TRAV3.                       04840066   *
   12              MULTIPLY TRAV3 BY TRAV1 GIVING ZMAXINT.                             04850066   *
   12              DIVIDE TOTALI (I6) BY SECI GIVING MBYTE.                            04860054   *
   12              SET I7 TO 1. SET I8 TO 1.                                           04870068   *
   12              MOVE VALEUR (I6) TO ZVAL.                                           04880043   *
   12              DIVIDE TOTB (I6) BY SECI GIVING MOYE.                               04890031   *
   12              MOVE '*' TO ASTL ASTW ASTX  ASTV ASTB.                              04900069   *
   12              WRITE ZIMP FROM ZIMP9.                                              04910061   *
   12              WRITE ZIMP FROM TIT10 . MOVE 3 TO CPTL.                             04920076   *
    8         LK. IF TRAV1 GREATER THAN BTHPI (I6, I7) GO TO FOL3.                    04930045   *
   12              MOVE CPTL TO PAGEDS (I8). GO TO F7.                                 04940071   *
    8         FOL3. MOVE 0 TO PAGEDS (I8).                                            04950066   *
    8         F7. SET I7 UP BY 1. SET I8 UP BY 1. ADD 1 TO CPTL.                      04960071   *
   12              IF I7 LESS THAN 9 GO TO LK.                                         04970070   *
   12              WRITE ZIMP FROM'TIT14. WRITE ZIMP FROM ZIMP10.                      04980070   *
   12              WRITE ZIMP FROM TIT14.                                              04990070   *
   12              SET I6 UP BY 1.                                                     05000031   *
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  MOD FLAGS
                                                                                 05010031    *
   12            IF I6 LESS THAN 19 GO TO PG.                                     05020065    *
    8      S7.  IF MEMT = 2 GO TO FIN.                                           05030032    *
   12           SET I1 TO 1. SET I3 TO 1. SET I6 TO 1.                           05040032    *
    8      KH.  MOVE 0 TO INUMMESS (I1).                                         05050032    *
   12           MOVE 0 TO ILONGMOY (I1).                                         05060032    *
   12           MOVE 0 TO INUMIN   (I1).                                         05070032    *
   12           MOVE 0 TO INUMOUT  (I1).                                         05080033    *
   12           SET I1 UP BY  1. IF I1 LESS THAN 302 GO TO KH.                   05090054    *
    8      Z2.  MOVE 0 TO IMESSAPL (I3).                                         05100032    *
   12           MOVE 0 TO ILONGAPL (I3).                                         05110032    *
   12           MOVE 0 TO IAPLIN   (I3).                                         05120032    *
   12           MOVE 0 TO IAPLOUT  (I3).                                         05130032    *
   12           SET I3 UP BY 1.                                                  05140032    *
   12           IF I3 LESS THAN 51 GO TO Z2.                                     05150032    *
    8      P7.  MOVE 0 TO MAXINT (I6).                                           05160032    *
   12           MOVE 0 TO TOTB (I6).                                             05170032    *
   12           SET I6 UP BY 1.                                                  05180032    *
   12           IF I6 LESS THAN 19 GO TO P7.                                     05190040    *
    8      ZER1.                                                                 05200009    *
   12           MOVE 0 TO LONGINT.                                               05210017    *
   12           MOVE 0 TO NUMINT.                                                05220011    *
    8      IF3. IF MEMT NOT = 2 GO TO READVAR.                                   05230009    *
    8      FIN. CLOSE FPRIM.                                                     05240015    *
   12           GOBACK.
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--

   1  ************************************

   1  DATE  HOUR MIN  SEC
   1   229 *    * 29 * 14

   1  ************************************

   1  DATE  HOUR MIN  SEC
   1   229 *    * 47 * 29

   1  FOR THIS INTERVAL

   1    LONG       NBRE       NBIN      NBO   TAUX    FREQ
   1   0010 *   1.090 *          * 1.090 * 0,36 * 0,99
   1   0013 *      13 *      13 *       * 0,00 * 0,01
   1   0015 *     154 *         *     154 * 0,05 * 0,14
   1   0019 *       3 *         *       3 * 0,00 * 0,00
   1   0021 *      84 *         *      84 * 0,02 * 0,07
   1   0022 *     100 *      35 *      65 * 0,03 * 0,09
   1   0023 *      33 *         *      33 * 0,01 * 0,03
```

```
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--
```

1	0271	1	1	0,00	0,00	
1	0272	6	6	0,00	0,00	
1	0274	1	1	0,00	0,00	
1	0279	1	1	0,00	0,00	
1	0288	1	1	0,00	0,00	
1	0291	1	1	0,00	0,00	
1	0299	2	2	0,00	0,00	
1	0301	100	100	0,03	0,09	

nom de l'application → · nombre de message → · longueur moyenne → · nbre de mes en entrée → · en sortie → · % des messages refusés → · fréquence →

2 BY APPLICATION RUNNING

1 NOM	NBRE	LONG		NBOUT	TAUX	FREQ
1 T901TS	115	85	42	73	0,03	0,10
1 T902TS	100	72	35	65	0,03	0,09
1 T904TS	93	29	32	61	0,03	0,08
1 T905TS	108	52	39	69	0,03	0,09
1 T906TS	122	49	46	76	0,04	0,11
1 T909TS	94	28	33	61	0,03	0,08
1 T908TS	92	29	32	60	0,03	0,08
1 T907TS	108	52	40	68	0,03	0,09
1 T910TS	86	47	31	55	0,02	0,07
1 T911TS	88	95	32	56	0,02	0,08
1 T915TS	107	46	40	67	0,03	0,09
1 T913TS	109	60	40	69	0,03	0,09
1 T912TS	99	46	36	63	0,03	0,09

START COL ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--

```
  2   NBRE      LONG    TAUX    FREQ
  1    3.000 *    53 * 2,73 * 1095

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    80 *    133 *   21584 * 03,77 *     573 *   628520
  2   NUMBER PAGES XI GOOD

 16                     6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    88 *    117 *   20000 * 03,68 *     590 *   646080
  2   NUMBER PAGES XI GOOD

 16                     6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    96 *    109 *   19656 * 03,56 *     599 *   656208
  2   NUMBER PAGES XI GOOD

 16                     6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1   104 *    100 *   18656 * 03,45 *     607 *   664928
  2   NUMBER PAGES XI GOOD
```

longueur moyenne

nbre de messages par seconde

nbre de messages à l'instant t_i dans l'intervalle

$S(x_i)$

longueur en seconde de l'intervalle

résumé sur l'intervalle

nbre moyen de tampons pris par seconde

$(O_{80}) * (t_m - t_1)$

x_i

(Max_{80})

(O_{80})

le x_i 80 est possible pour un nombre de pages à partir de 6.

START COL ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9-

```
 12              5   6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1   112 *     91 *    18032 * 03,37 *    621 *   680064
  2   NUMBER PAGES XI GOOD

 12              5   6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1   120 *     80 *    17088 * 03,29 *    631 *   691776
  2   NUMBER PAGES XI GOOD

 12              5   6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1   128 *     80 *    17600 * 03,24 *    649 *   711400
  2   NUMBER PAGES XI GOOD

 12              5   6   7   8   9  10

  1     BSZ    USED    MAX      E(USED)    A(L)      TOT
  1   136 *     75 *    17472 * 03,21 *    667 *   731120
  2   NUMBER PAGES XI GOOD
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--
   12             5   6   7   8   9  10

    1     BSZ   USED   MAX      E(USED)   A(L)      TOT
    1   144 *    69 *   16848 * 03,17 *   686 *   751248
    2   NUMBER PAGES XI GOOD

   12             5   6   7   8   9  10

    1     BSZ   USED   MAX      E(USED)   A(L)      TOT
    1   152 *    65 *   16576 * 03,14 *   704 *   771456
    2   NUMBER PAGES XI GOOD

   12             5   6   7   8   9  10

    1     BSZ   USED   MAX      E(USED)   A(L)      TOT
    1   160 *    62 *   16704 * 03,12 *   724 *   793672
    2   NUMBER PAGES XI GOOD

   12             5   6   7   8   9  10

    1     BSZ   USED   MAX      E(USED)   A(L)      TOT
    1   168 *    62 *   17280 * 03,10 *   746 *   817200
    2   NUMBER PAGES XI GOOD

   12             5   6   7   8   9  10
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--

  1      BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    176 *     61 *    17608 * 03,08 *     764 *   837496
  2    NUMBER PAGES XI GOOD

 12              5    6    7    8    9   10

  1      BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    184 *     54 *    18176 * 03,06 *     784 *   859392
  2    NUMBER PAGES XI GOOD

 12              5    6    7    8    9   10

  1      BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    192 *     55 *    18480 * 03,05 *     805 *   882552
  2    NUMBER PAGES XI GOOD

 12              5    6    7    8    9   10

  1      BSZ    USED    MAX      E(USED)    A(L)      TOT
  1    200 *     54 *    19040 * 03,03 *     825 *   904128
  2    NUMBER PAGES XI GOOD

 12              5    6    7    8    9   10
```

START COL						
1	T918TS	76	97	27	49	0,03 * 0,11
1	T917TS	56	61	20	36	0,02 * 0,08
1	T920TS	58	29	20	38	0,02 * 0,09
1	T921TS	50	67	18	32	0,02 * 0,07
1	T919TS	64	41	24	40	0,02 * 0,10
1	T922TS	62	44	23	39	0,02 * 0,09
1	T923TS	69	43	25	44	0,03 * 0,10
1	T924TS	57	26	20	37	0,02 * 0,08
1	T926TS	69	40	26	43	0,03 * 0,10
1	T925TS	71	40	25	46	0,03 * 0,11
1	T927TS	74	91	27	47	0,03 * 0,11
1	T930TS	64	52	23	41	0,02 * 0,10
1	T928TS	60	27	22	38	0,02 * 0,09
1	T929TS	64	31	23	41	0,02 * 0,10
1	T931TS	76	53	28	48	0,03 * 0,11
1	T933TS	46	52	16	30	0,02 * 0,07
1	T932TS	68	30	24	44	0,03 * 0,10
1	T934TS	45	42	16	29	0,02 * 0,07
1	T936TS	60	57	22	38	0,02 * 0,09
1	T935TS	73	89	26	47	0,03 * 0,11

1 RESUMED OF THE VALUES OBSERVED

2 NBRE	LONG	TAUX	FREQ
1 2.195	48	3,46	634

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1     80 *     9 *   22040 * 04,50 *   684 *  434112
   2   NUMBER PAGES XI GOOD

  16                  6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1     88 *     9 *   20800 * 04,43 *   708 *  449440
   2   NUMBER PAGES XI GOOD

  16                  6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1     96 *     9 *   20496 * 04,29 *   720 *  456960
   2   NUMBER PAGES XI GOOD

  16                  6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1    104 *     9 *   20064 * 04,16 *   733 *  465168
   2   NUMBER PAGES XI GOOD

  16                  6   7   8   9  10
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--

   1      BSZ    USED    MAX      E(USED)    A(L)      TOT
   1    112 *     10 *    19872 * 04,08 *    752 *   476928
   2    NUMBER PAGES XI GOOD

  16                    6    7    8    9   10


   1      BSZ    USED    MAX      E(USED)    A(L)      TOT
   1    120 *      8 *    18048 * 04,00 *    769 *   487680
   2    NUMBER PAGES XI GOOD

  12                    5    6    7    8    9   10


   1      BSZ    USED    MAX      E(USED)    A(L)      TOT
   1    128 *      8 *    18800 * 03,95 *    791 *   501600
   2    NUMBER PAGES XI GOOD

  12                    5    6    7    8    9   10


   1      BSZ    USED    MAX      E(USED)    A(L)      TOT
   1    136 *      7 *    19136 * 03,92 *    815 *   517296
   2    NUMBER PAGES XI GOOD

  16                    6    7    8    9   10
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--
   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1    144 *     8 *   18792 * 03,88 *   838 *   531360
   2    NUMBER PAGES XI GOOD

  12              5   6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1    152 *     8 *   19040 * 03,85 *   864 *   547904
   2    NUMBER PAGES XI GOOD

  12              5   6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1    160 *     8 *   19256 * 03,83 *   889 *   563760
   2    NUMBER PAGES XI GOOD

  16                  6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
   1    168 *     8 *   20640 * 03,81 *   915 *   580560
   2    NUMBER PAGES XI GOOD

  16                  6   7   8   9  10

   1      BSZ   USED   MAX      E(USED)   A(L)      TOT
```